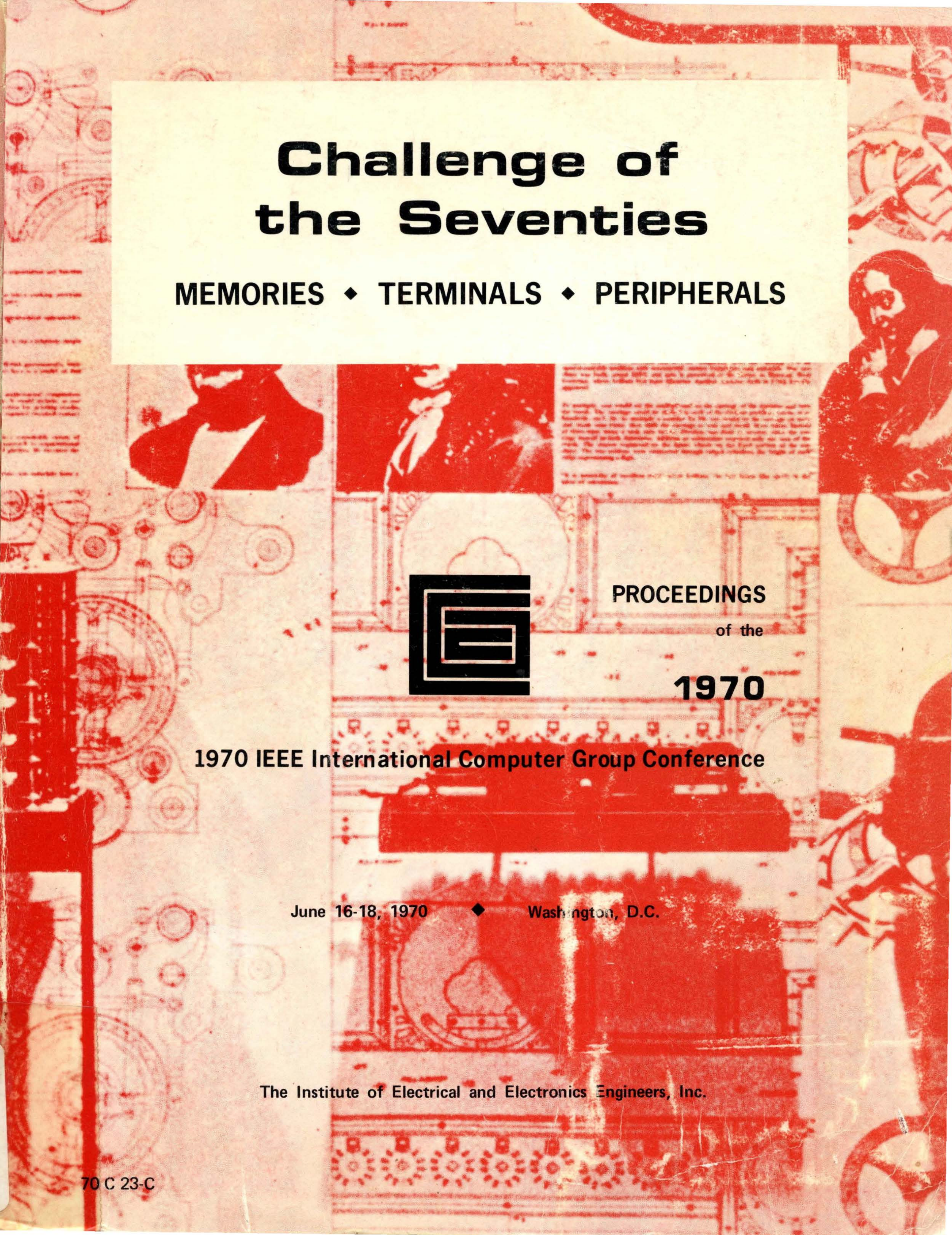
PROCEEDINGS of the 1970 IEEE International Computer Group Conference

TK
7885
A1I18
1970



Challenge of the Seventies

MEMORIES ◆ TERMINALS ◆ PERIPHERALS

PROCEEDINGS
of the
1970

1970 IEEE International Computer Group Conference

June 16-18, 1970 ◆ Washington, D.C.

The Institute of Electrical and Electronics Engineers, Inc.

70 C 23-C

MEMORIES

TERMINALS

PERIPHERALS

PROCEEDINGS

of the

1970 IEEE International Computer Group Conference

June 16-18, 1970

Washington Hilton Hotel

Washington, D.C.

IEEE COMPUTER GROUP
ADMINISTRATIVE COMMITTEE

Officers

Chairman, E. J. McCluskey, Stanford University
Vice Chairman,
  Chapter Activities: Sei Shohara, Xerox Data Systems
Vice Chairman,
  Technical Activities: Dr. A. S. Hoagland, IBM Corporation
Secretary: Thomas E. Lindsay, DIVA Assoc. Inc.
Treasurer: Howard Zeidler, Stanford Research Institute
Jr. Past Chairman: L. C. Hobbs, Hobbs Associates

Term Expires 1970

David R. Brown, Stanford Research Institute
Dr. Curt F. Fey
Dr. A. S. Hoagland, IBM Corporation
Thomas E. Lindsay, DIVA Assoc. Inc.
Milton A. Lipton, U. S. Army Electronics Command
Sam Nissim, Electronic Arrays, Inc.
Ralph J. Preiss, IBM Corporation
Rex Rice, Fairchild Corporation
Sei Shohara, Xerox Data Systems
John H. Worthington, IBM Corporation

Term Expires 1971

R. B. Arndt, Univac Division Sperry Rand
E. Blumenthal, Burroughs Corporation
Dr. Bruce Briley, Bell Telephone Laboratories
T. R. Finch, Bell Telephone Laboratories
Dr. A. L. Hopkins, MIT
Dr. Harry D. Huskey, University of California
Dr. Robert A. Kudlich, Raytheon Company
C. W. Rosenthal, Bell Telephone Labs
Dr. S. S. Yau, Northwestern University
Howard Zeidler, Stanford Research Institute

Technical Committees Chairmen

Computers and Communications, J. J. Eachus
Computer Elements, G. S. Mitchell
Computer Systems, J. L. Maddox
Data Acquisition and Control, A. L. Hopkins
Design Automation, H. Freitag
Switching & Automata Theory, R. E. Miller
Management Data Processing Applications,
  G. W. Patterson
Programming, G. J. Moshos
Pattern Recognition, C. K. Chow
Peripheral Equipment, C. R. Fisher
Fault-Tolerant Computing, A. Avizienis

Standing Committees Chairmen

Technical Activities, A. S. Hoagland
Chapter Activities, S. Shohara
  Area Committees - Midwest, D. B. Simmons;
    Mideastern, N. R. Kornfield; West Coast,
    L. D. Amdahl
Awards, W. L. Anderson
Chapter Newsletter, R. A. C. Lane
Computer Standards, D. R. Brown
Conferences and Meetings, S. Yau
Constitution and Bylaws, J. H. Worthington
Distinguished Visitors Program, N. Prywes
Fellows, N. Rochester
Membership, R. B. Arndt
Nominations, S. Levine
Publications, M. H. Lewin
Social Implications of Computers, H. T. Larson

AFIPS Directors for IEEE

L. C. Hobbs   R. A. Kudlich   E. J. McCluskey

Editorial Board
H. D. Huskey, Editor-in-Chief
J. Kirkley, Editor/Computer Group News

Copyright 1970 by The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th St., New York, New York 10017

Price $5.00 IEEE Members; $7.00 Nonmembers

IEEE Catalog Number 70-C-23-C

Available from: IEEE, 345 East 47th St., New York, New York 10017

CONTENTS

Conference Chairman's Message viii

Technical Program Chairman's Message viii

SESSION 1: TOMORROW'S TECHNOLOGY AND MARKETS
Chairman: Robert B. Forest, DATAMATION

Tomorrow's Technology 1
L. D. Amdahl, Compata, Inc.

Tomorrow's Technology and Markets 2
J. Peter Ross, Quantum Science Corporation

Future Technical and Market Trends for Computer Peripherals 5
L. C. Hobbs, Hobbs Associates

Software in the Seventies 7
Werner L. Frank, Informatics, Inc.

Memories - Technology and Markets for the 70's 9
George R. Cogar, Cogar Corporation

Digital IC Market and Technology Trends 10
Robert B. Seeds, Fairchild Camera and Instrument Corporation

SESSION 2: PACKAGING FOR ELECTRONIC SYSTEMS
Chairman: Robert E. McMahon, Lincoln Laboratories, MIT

Semiconductor Chip Interfacing 13
T. J. Matcovich, Drexel University

Ceramics for Packaging 20
D. L. Wilcox, Components Division, IBM Corporation

Tradeoffs in Packaging Beam Lead Integrated Circuit Systems (Abstract) 30
M. T. Skubiak and D. J. Wadsworth, Bell Telephone Laboratories, Inc.

Thermal Design Optimizations in Large Digital Computers (Abstract) 30
R. C. Chu, J. H. Seely, V. W. Antonetti and A. L. Pascuzzo, Systems Development Division, IBM Corporation

Packaging Aspects of the UNIVAC 9200 Design 31
W. F. Simon, Univac Division of Sperry Rand Corporation

SESSION 3: COMPUTING POWER IN TERMINALS
Chairman: Prof. Edward L. Glaser, Andrew Jennings
Computing Center, Case Western Reserve University
Vice Chairman: Prof. Niel E. Wiseman, University Mathematical
Laboratory, Cambridge University, England

How Much Logic Should Be Put in a Display Terminal? 37
Ivan E. Sutherland, Evans & Sutherland Computer Corporation

The Use of Video Technology To Provide Economically Attractive General-Purpose Graphics . . 38
T. O. Ellis, The RAND Corporation

Consoles: Intelligence at the Man Site 39
Glen Culler, Culler-Harrison Company

The Complexity of General-Purpose Remote Computer Terminals 40
Joseph F. Ossanna, Bell Telephone Laboratories, Inc.

SESSION 4: NEW MEMORY DEVICES
Chairman: Richard O. McCary, Research and Development Center, General Electric Company

High Bit Density, Closed Flux-Path Magnetic Thin Film Memory - Fine Striped Memory 43
S. Oshima, T. Kobayashi, K. Amano, T. Kamibayashi, A. Kada, Y. Komazawa and K. Komuro, Kokusai Denshin Denwa Co., Ltd., Tokyo, Japan

A Proposed Fast-Access Large-Capacity Ferroelectric Memory 56
D. W. Chapman, Systems Development Division, IBM Corporation

Soniscan - A Sonically Accessed Magnetic-Film Memory 64
H. Rubenstein, R. Hornreich and J. Teixeira, Applied Research Laboratory, Sylvania Electronic Systems

The DYNABIT^{T} Device: A Versatile New Memory 73
B. D. Broadbent, Advanced Technology Department, Hughes Aircraft Company

Holographic Memories 83
P. Graf, Siemens A. G., Research Laboratory, Munich, Germany

SESSION 5: HIGH RISK MONEY FOR NEW COMPUTER COMPANIES: HOW, WHEN AND WHERE
Chairman: Robert F. Johnston, Johnston Associates

Venture Capital 89
Robert F. Johnston, Johnston Associates

SESSION 6: MAIN FRAME MEMORY TECHNOLOGY: A DEBATE
Chairman: Tudor R. Finch, Bell Telephone Laboratories, Inc.
Vice Chairman: Georges Grunberg, COFELEC, Choisy-Le-Roi, France

Main Frame Memory Technology - A Debate 93
T. R. Finch, Bell Telephone Laboratories, Inc.

Magnetics vs. Semi-Conductors as the Memory Elements of the Future 94
Leo F. Slattery, Memory Products Division, Fabri-Tek, Inc.

Random Access Semiconductor Memories 96
David A. Hodges, Bell Telephone Laboratories, Inc.

Core Memories 100
P. A. Harding, Electronic Memories and Magnetics

Random Access Memories - Plated Wire or Ferrite Core? 103
G. A. Fedde, Univac Division of Sperry Rand Corporation

Multiple Technology Hybrid Semiconductor Mainframe Memory 105
Wally Raisanen, Integrated Circuit Center, Motorola Inc., SPD

In Defense of the Single Technology Approach to Semiconductor Main Frame Memories for the '70's 107
Gordon E. Moore, Intel Corporation

SESSION 7: TERMINALS FOR NEWS PROCESSING AND EDITING
Chairman: Prof. J. Francis Reintjes, Department of Electrical Engineers, MIT

Computer Terminals for Newspapers: Applications and Requirements 109
Richard S. Marcus, Electronic Systems Laboratories, MIT, and Jules Tewlow, Research Institute, American Newspaper Publishers' Association

The Wire Service and the Video Display 115
James F. Darr, Communications Department, United Press International

Publishing Systems Development 120
R. F. Jefferson, Federal Systems Division, IBM Corporation

Low-Cost Source Data Terminal for Technical Text 126
Loren P. Meissner, Mansfield L. Clinnick and Robert A. Belshe, Lawrence Radiation Laboratory, University of California, Berkeley

SESSION 8: SPECIAL PURPOSE TERMINALS AND PERIPHERALS
Chairman: Dr. Wolfgang Giloi, Technical University of Berlin, Germany

Design Considerations for a Remote Graphic Computer Terminal. 141
Donald C. Austin and C. Norman Winningstad, Tektronix, Inc.

Hard Copy from a Direct View Storage Tube Remote Display Terminal 148
A. Vezza, Project MAC, MIT

A Fast Contactless Optoelectronic Printer 159
J. G. Berny, J. P. Bourgoin and A. A. Donzel, Laboratoires de Marcoussis, Compagnie Generale d'Electricite

A Hybrid Special Purpose Computer for High Speed FFT .165
W. Giloi, Technical University of Berlin, Germany

SESSION 9: LOW COST TELEDISPLAYS
Chairman: Charles R. Fisher, Electronic Engineering Company

Monoscope Character Generation 171
J. H. Selander, Communications and Data Processing Operation, Raytheon Company

LSI Refresh Memories 175
Enrique Moreno, Beehive Medical Electronics, Inc.

Graphic Displays Using Image Storage Techniques 176
Thomas B. Cheek, Computer Displays, Inc.

Centralized Store Refresh Memories - A Provincial View 178
N. Clark, TEC, Inc.

Engineering Design of UNISCOPE 100 Terminal 180
K. R. Fechter, T. M. Soller and M. L. Granberg, Communications and Terminals Division, Univac

Flexibility in Low-Cost Interactive Computer Terminals 186
Donald R. Haring, Computek, Inc.

SESSION 10: SOFTWARE AND TERMINALS
Chairman: Henry S. McDonald, Bell Telephone Laboratories, Inc.

Data Base Input and Text Handling in an OCR System 189
Gerhard Reitz, MACRODATA Company

The Use of APL\360 in Mechanical Analysis 195
Alan L. Jones, Systems Development Division, IBM Corporation

An Expanded Terminal Design for Clinical Medical Research 205
H. William Perlis, Steven E. Winson and Eugene M. Strand, University of Alabama Medical Center

SESSION 11: DIAGNOSABLE DESIGN AND DIAGNOSIS
Chairman: Ralph J. Preiss, Systems Development Division, IBM Corporation

Reliability of Integrated Circuits 213
R. J. Sahni, Technology Division, Cogar Corporation

Alternating Logic and Its Application to Fault Corporation 220
H. Yamamoto, T. Watanabe and Y. Urano, Research and Development Laboratory, Kokusai Denshin Denwa Co., Ltd.

Error-Free Decoding for Failure-Tolerant Memories 229
W. C. Carter, D. C. Jessep and A. Wadia, Thomas J. Watson Research Center, IBM Corporation

A Method of Deriving Minimal Complete Sets of Text-Input Sequences Using Boolean Differences 240
P. N. Marinos, Department of Electrical Engineering, Duke University

An Algorithm To Compute a Test To Distinguish between Two Failures in a Logic Circuit . . .247
J. Paul Roth, Thomas J. Watson Research Center, IBM Corporation

A Path Sensitizing Algorithm for Diagnosis of Binary Sequential Logic 250
Thomas A. Kriz, Automation Systems Development Department, Federal Systems Division, IBM Corporation

SESSION 12: DIG DEEPER SESSION: MAIN FRAME MEMORY TECHNOLOGY
(Continuation of SESSION 6)
Chairman: Georges Grunberg, COFELEC
Vice Chairman: Tudor R. Finch, Bell Telephone Laboratories, Inc.

SESSION 13: ADVANCED TERMINALS AND SENSOR ACTUATOR DEVICES
Chairman: Prof. Jerome A. Feldman, Computer Science Department, Stanford University

Electroluminescent Flat Panel Display 261
M. Yoshiyama, H. Kawarada and T. Sato, Wireless Research Laboratory, Matsushita Electric Industrial Co., Ltd., Osaka, Japan

Head-Mounted Display Terminal 270
D. L. Vickers, Computer Science, University of Utah

Eye Tracking as a Computer Input Technique 278
Sam L. Geffner, Project MAC, MIT

Speech Input Terminals for Computers: Problems and Prospects 282
D. R. Reedy, Computer Science Department, Carnegie-Mellon University

Manipulators as Terminal Devices 290
Alan C. Kay, Stanford University

SESSION 14: FUNCTIONAL AND CONTROL MEMORIES
Chairman: James L. Previte, Rome Air Development Center

Processing Memories 299
J. P. Bartlett, Fairchild Camera and Instrument Corporation

The Use of Read Only Storage Modules To Perform Complex Logic Functions 307
Joel C. Leininger, Components Division, IBM Corporation

Functional Memory as a General Purpose Systems Technology 314
M. Flinders, P. L. Gardner, R. J. Llewelyn and J. F. Minshull, IBM United Kingdom Laboratories, Ltd.

A Modular Plated-Wire Associative Processor 325
L. C. Fulmer and W. C. Meilander, Goodyear Aerospace Corporation

An Associative Memory for Executive Control Functions in an Advanced Avionics Computer System 336
Robert O. Berg and Melvin D. Johnson, Honeywell Systems and Research Center

SESSION 15: ROTATING MEMORIES AND ELECTRONIC DRUMS - PERFORMANCE AND COST CONSIDERATIONS
Chairman and Panel Leader: Edwin J. Blumenthal, Defense, Space & Special Systems Group, Burroughs Corporation

Session Summary: Rotating Memories and Electronic Drums - Performance and Cost Considerations 343
Edwin I. Blumenthal, Defense, Space and Special Systems Group, Burroughs Corporation

Storage Improvements and Systems' Cost/Performance 345
R. F. Arnold, IBM Corporation

DYNABIT^{T} Mass Memory 352
B. D. Broadbent, Advanced Technology Department, Hughes Aircraft Company

Semiconductor I/O Storage 354
R. H. F. Lloyd, Advanced Memory Systems, Inc.

A Brief Review of Current Design Trends and Aspirations in Electromechanical Mass Storage 355
R. Tickell, Univac Division of Sperry Rand Corporation

Head Per Track Disk File Mass Memories 357
R. W. Jack, Westlake Plant, Burroughs Corporation

Moveable Head, Fixed Disc Systems 360
Irving L. Wieselman, Data Products Corporation

Design Directions for Rotating Mass Storage 363
Harold S. Yang, Information Storage Systems

SESSION 16: SYSTEM ARCHITECTURE
Chairman and Panel Leader: Michael J. Flynn, Northwestern University
Vice Chairman and Panel Leader: Dr. Smil Ruhman, The Weizmann Institute of Science, Rehovoth, Israel

Parallelism in Systems Architecture 367
D. L. Slotnick, Department of Computer Science, University of Illinois

Storage and I/O Parameters and Systems Potential 371
G. M. Amdahl, IBM Corporation

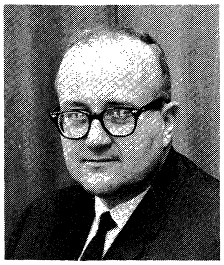
Communications and System Architecture 373
L. G. Roberts, Advanced Research Projects Agency

Systems Aspects of Storage 374
J. E. Thornton, Advanced Design Laboratory, Control Data Corporation

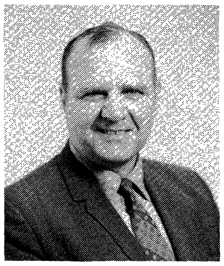
AUTHOR INDEX 376

EXHIBITORS 377

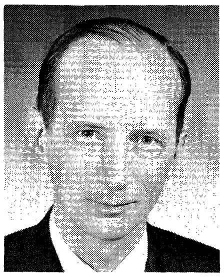
STEERING COMMITTEE 380



Chairman
Bob O. Evans
IBM Corporation



Vice Chairman
Donald D. Doll
IBM Corporation



Technical Program Chairman
Dr. Gardiner L. Tucker
Assistant Secretary of Defense
for Systems Analysis
Department of Defense

Co-Chairman
Dr. Martin Greenberger
The Johns Hopkins University

Co-Chairman-International
Mr. Maurice Allegre
Office of the Prime Minister
Republic of France

MEMORY SESSIONS

Director
Gordon S. Mitchell
National Security Agency

Vice Director
Tudor R. Finch
Bell Telephone Laboratories

TERMINAL/PERIPHERAL SESSIONS

Director
Michael Dertouzos
Massachusetts Institute of Technology

Vice Director
Edwin Blumenthal
Burroughs Corporation

CONFERENCE CHAIRMAN'S MESSAGE

Curiosity about new developments is a hallmark of professionalism in the technical community. This year's IEEE Computer Group Conference is designed to provide an opportunity for professionals to hear intensive reviews of several dynamic areas of computer technology: memories, terminals and peripherals. Rapid technology changes, coupled with the explosive growth in applications of memory, terminal and peripheral technology, were fundamental factors in planning the program, which is set in the framework of an industry barely twenty years old, embarking on its third and most challenging decade. To address the decade ahead, we have succeeded in committing internationally known leaders in research, development and application to our podium. The participation of more than 70 of our industry's leading companies in our international exhibit provides an effective counterpoint to the technical sessions.

Bob O. Evans

TECHNICAL PROGRAM CHAIRMAN'S MESSAGE

Computer technology has enjoyed two decades of revolutionary development, and an explosive market has given new technologies wide acceptance. The present decade will witness some important changes. Among them, a rapidly expanding international computer industry, with a larger share of the market occupied in peripherals and memories and an enabling development of new peripheral and memory technologies.

This Committee has carefully selected a distinguished group of participants to illuminate the pivotal technical issues facing the international engineering and design community in memory and peripheral system design. We have added two plenary sessions to develop for you an understanding of the marketplace of the 70's and to examine the systems architecture implication of the memory and peripheral developments. We have been highly selective and have accepted from the broad range of papers submitted only those whose relevance to our central theme was clear and whose quality was excellent.

You will find this Conference an important educational experience; one which, with your enthusiastic participation, should be a creative contribution to the industry.

Gardiner L. Tucker

TOMORROW'S TECHNOLOGY

L. D. Amdahl, Compata, Inc.

There are many facets of technology that pace the computer industry: components, processor architecture, and system architecture. In the early years of computers, electronic component technology was the undisputed motivating force in major changes. Magnetic core memories and the replacement of vacuum tubes with semiconductors are evidence of this. However, advances in component technology are being challenged in importance by other factors. For example, in going from the second generation to the third generation of computers, IBM's decision to amalgamate its business and scientific product lines seems more significant than its concomitant aspects of 8-bit bytes, hybrid circuits, operating systems, and ROS microprogramming. Extrapolating into the mid-seventies, we can expect that technology improvements will be concentrated on serving the user on his terms. This will include a strong trend to place I/O devices near him (from our limited perspective this means remote terminals). It will also permit him to use his computer system directly with his data and in his language.

There are some general trends in technology that are worthy of note that have been observed on a statistical basis, and appear to be rather reliable without specific knowledge of innovations. One such trend is in the performance-to-price ratio. Narrowly applied to central processors, this ratio improves from 40% to 70% per year as averaged over new equipment in each succeeding year. Stated another way, price/performance doubles or triples every two years. Of course, we realize that a total computer environment, including support personnel, cannot typically realize more than a fraction of this improvement.

For a general class of equipment, the performance trend to greater speed has typically been greater than the price reduction trend. However, at the low end of the computer spectrum--the minicomputer --this trend has been dominant. What we classify as a general purpose computer decreases in price by 30% to 35% per year. Thus the cheap computer of 1968 was priced like a Cadillac, today's computer is priced like a Chevy, and one expects that there will be a computer available at the price of a Volkswagen in 1972. This rapid change in price continually exposes new applications in which g.p. computers are useful.

Specifically in component technology, it appears that the dominating influence on the design of digital equipment during the next few years will come from semiconductors. LSI components will affect both memories and processors. For example, small memories of a few hundred bytes have generally been unavailable at an attractive price in the past. This is beginning to change, and will have strong implications in the design of peripheral controllers, remote terminals, desk calculators and the like. High speed buffer memories like that of IBM's cache memory used in the 360/85 is expected to be important in memory hierarchies of medium and large scale computers.

LSI is also expected to usher in an era of "cheap logic" for implementation of control functions. While it is premature to predict a general onslaught of a "computer-on-a-chip," the trend to LSI by way of MSI will lead to continuing changes in hardware/software tradeoffs. A larger spectrum of computers can be expected to evolve, with frugal design leading to very low cost machines and performance-oriented design leading to the imbedding of portions of present operating systems in special hardware.

The emphasis on electromechanical components is expected to change in the coming years. Peripherals used in remote batch terminals will have a greater emphasis on price than on performance. Disc memory systems are expected to evolve to higher performance through higher recording densities and greater emphasis on head-per-track organization. Interestingly, relatively low capacity disc memories may grow in use despite the advances of static memory technology because of the volatility of semiconductor memories.

The technology of computer architecture, reacting both to new components and to user needs, will emphasize microprogramming and inevitably, firmware. This will help in achieving all kinds of compatibility -- upward, downward and backward. The system dependability required by massive on-line use of computers can be expected to result in substantial equipment redundancy. This in turn will require the sophisticated failure detection and system recovery techniques that, in the past, have been employed only in complex-time systems.

In conclusion, technology will set an exciting pace for all of us in the 70's. As a result of continuing technical innovations, the body of computer users will be extended far more deeply into our society.

TOMORROW'S TECHNOLOGY AND MARKETS

J. Peter Ross, Quantum Science Corporation

INTRODUCTION

The markets, products, technologies and companies of the 70's will be affected and controlled by three major trends:

- The rapid introduction of dispersed systems that will dictate new product development, particularly in the areas of terminals, communications equipment, memories, CPU architecture and organizations, and application software.
- The accelerating movement towards the total service concept -- brought about by the increasing complexity of the computer environment of software and operations adding to the cost of data processing.
- The growing importance of foreign markets, which are increasing 50% faster than those in the U.S., and the competition from gigantic foreign companies preparing to attack world markets.

Other members of this panel will discuss their viewpoints of the trends in the 70's in technology, software, memories, and peripherals. I will discuss the underlying market considerations, the size of these markets for the next half decade and highlight some opportunities and pitfalls.

REMOTE COMPUTING - THE MAJOR HARDWARE DETERMINANT OF THE 70's

Dispersed systems will dominate the domestic market in the 70's. By 1975 the service (network information service) part of this market will provide $2 billion in revenues while the in-house (dedicated information services) part will account for upwards of $6 billion of revenue per year. Thus, dispersed systems, which are negligible in size now, will account for 40% of computer industry revenues in 1975. The implications of these figures are enormous when you consider the magnitude of existing field inventories (over $40 billion worldwide) that must be replaced or revised to bring about this change.

- Dispersed systems will tend to evolve towards more and more intelligent job-oriented terminals and small processors at the periphery of the enterprise or computer network as advances in MOS and LSI technology reduce the costs of circuitry and memories.
- Central processors will be optimized by architecture organizations and software to meet the needs of multiple users, including conversational time-sharing, remote batch processing and on-line inquiry -- all on the same network.
- Systems will include immense mass memories, originally of the disk file type, but giving way eventually to advances in MOS and LSI technology, which will provide faster-access memories at lower prices.
- Communications costs will be reduced by the implementation of data microwave networks, advances in methods of compressing data and the evolution of systems that process more data at the remote terminal site.
- The total service concept will increase in importance for both the in-house and network-service approaches to remote computing. Broad competence, marketing orientation,

strong organizations and financial resources will be more important than specific bits of technical excellence as a basis of leadership.

COMPUTER SERVICES - THE FASTEST GROWING SECTOR

Over the next 4 years domestic hardware shipments will grow at a rate of 17% from $5.2 billion in 1969 to $9.9 billion in 1973. During the same period, computer services will grow from $7.9 to $20.1 billion, a compound growth rate of 26%.

Thus, computer software and operations -- which already take two thirds of the data processing user's budget -- will have an accelerated growth rate because:

- * Computational costs are declining as computer power increases. The next generation of hardware is expected to have a 3:1 increase in price performance.
- * Computer applications are becoming more complex as dispersed systems are implemented and increasingly difficult jobs are attacked.
- * Independent service companies are increasing their share of the market as manufacturers unbundle and users find themselves unable to cope with the growing complexities of hardware selection, systems design, programming and operations.

The largest share of the independent service market will be captured by vendors who offer a total service to the users -- hardware, software and operations. There are three ways this service can be provided.

- * Service bureaus, where the customer brings data to be processed, presently have revenues exceeding $750 million per year. By 1973 revenues will increase to $1.4 billion.
- * Time-sharing, where data is transmitted from a terminal in the user's office to a central facility to be processed, presently have revenues of $350 million. By 1973 these revenues will be $1.3 billion.
- * Facilities management, where the vendor supplies the total information service at a fixed price based on a long-range contract, is the fastest growing sector. From a base of $300 million this year, 1973 revenues will exceed $1.4 billion. This shows a compound growth rate of 63% per year.

Software expenditures are growing almost 50% faster than those for hardware and will exceed hardware expenditures by 1973.

- * The present value of computer user program libraries is $8 billion, with a current annual expenditure of $4 billion on software. By 1973 $11 billion a year will be spent on software and total value of the inventory will have increased to $25 billion.
- * The independent software supplier, however, will impact less than 10% of this total market since users will continue to spend 75% of the software dollar in-house and manufacturers will continue to supply certain software packages with their systems.
- * Of the estimated $1 billion market for independent software services, 25% will be in proprietary programs; this will play an important part in holding down total user costs.

Lack of trained people will be the restricting element in the growth of many companies and will provide a significant new business opportunity.

- * 900,000 people, not counting manufacturers' personnel, are presently concerned with computers, programming or operations. By 1973 this number will grow to 1.7 million plus an additional 500,000 manufacturers' personnel to bring total labor force in the data processing industry from 2.2 to 2.5 million people or 3% of the total U.S. work force.
- * The education of these people will become a major industry with revenues to the independents in excess of $150 million per year.

Other important sectors of the total market are shown on the enclosed chart.

The growth of this industry will be characterized by the startup of thousands of small companies, an active phase of merger and acquisition, and the emergence of three groups of companies.

- * 5 to 10 giant broad-based service and supply companies with revenues of over $200 million each.
- * 20-50 smaller companies averaging $50 million in sales, specializing on an industry or application basis.
- * 2,000 companies of under $10 million in revenue who will occupy less than 50% of the total market.

FOREIGN MARKETS - AND COMPETITION

Foreign markets and foreign competition will have an increasing importance in the success of the U.S. computer industry. In 1969, 39% of total U.S.-based computer companies' sales were made overseas. By 1973, 44% of sales will be to foreign markets. Foreign equipment sales will have a compound growth rate of 24% vs. 17% in the U.S. market.

Foreign markets are typically more profitable than domestic ones. If IBM had not had their profit from world trade in 1969 their net profits would have shown a decline over 1968.

The size, profitability, and importance of foreign markets is realized by foreign manufacturers. There are major efforts being mounted by concerns such as Seimens, Philips, ICC and Hitachi to impact world markets -- including the United States.

If the size of these companies and their potential as competitors is not familiar to you, consider that Philips, Hitachi and Seimens all have sales in excess of $2 billion a year and employ well over 675 thousand people.

These companies have the technical, financial and marketing strengths necessary to effectively impact world markets. In addition, they have the backing of their governments, who will not stand by while U.S. manufacturers carry away the spoils of their markets.

The ability of the Japanese to mass-produce and mass-market electronic equipment is evident in the revolution that brought about miniature radios, hi-fi equipment and portable television. The growth of the terminal market -- to a $2 billion business in 1973 -- will have an immense appeal to Japanese companies with electronics and CRT technical capabilities, and they will apply the marketing push that has seen them dominate foreign markets for cameras, calculators, and home entertainment equipment.

CONCLUSIONS

The challenges and opportunities of the 70's will be successfully met by companies who are able to:

- * Design and market products that will fill the needs of the rapidly expanding dispersed systems.
- * Provide a total service to solve the increasingly complex problems that plague the computer user and thus impact the total market potential of hardware, software, communications and operations.
- * Establish a marketing and support force that can operate on a multinational basis to impact lucrative foreign markets and defend home markets from foreign competitors.

Market Growth
(Domestic)

	1969	1973+	Compounded Growth Rate (%)
	--($ billions)--		
Software*	4.37	11.10	26%
Hardware shpts.	5.22	9.90	19%
Supplies	0.90	1.50	14%
Leasing (independent Co's.)	0.50	1.50	32%
Service bureaus	0.62	1.40	23%
Facility mgmt.	0.20	1.40	63%
Network information services	0.23	1.30	54%
Maintenance market*	0.42	0.86	20%
Used computers	0.08	0.60	65%
Education mkt.*	0.26	0.45	15%

+ Ranked by order of 1973 size

* Captive user & manufacturer markets included

FUTURE TECHNICAL AND MARKET TRENDS FOR COMPUTER PERIPHERALS

L. C. Hobbs, Hobbs Associates

NEED FOR IMPROVEMENT

Peripheral equipments will be one of, if not the major, limiting factor on computer systems performance and economics during the next decade. Continued improvements in semiconductor technology, which will provide LSI arrays for both logic and memory, and improved memory technologies, including both semiconductors and advanced magnetics, will lead to significant improvements in both performance and cost of the central computer and its internal memory. The cost of the central processor, already a small part of the total cost of computer systems ownership, is decreasing rapidly. Software and operating costs are assuming larger roles, but peripheral equipments are rapidly dominating the systems hardware costs. This is very obvious already in the minicomputer field where a single tape unit costs as much as a computer with 4,000 words of internal memory, and a disc unit may cost two to three times as much as the computer.

NEW APPROACHES

New approaches and new technologies for peripheral equipments are urgently needed to provide higher performance and lower cost than is likely to be achieved by gradual evolution of mass memories and input/output equipments presently in use. Although it is highly unlikely that mass storage and other peripheral equipments will be completely divorced from electromechanical components and techniques in the foreseeable future, this should be an important objective in planning future developments in peripheral equipments. The mechanical motions and components inherent in most present day peripheral equipments limit the speed and reliability and prevent order of magnitude improvements in cost.

Input/output approaches such as character recognition, voice recognition, and voice output will eventually provide improvements since low cost logic and storage provided by LSI semiconductor arrays will help make these techniques economically attractive and more widely used. Non-mechanical keyboards and non-impact printers also offer promise for major improvements in these types of peripheral equipments. Optical memory techniques have long range promise for partially replacing both magnetic tape equipment and magnetic disc files with devices that involve the deflection of laser beams rather than the mechanical motion of the transport. However, major improvements in lasers, and in particular laser deflection techniques, are required before optical storage can compete with magnetic surface storage for read/write storage. The development of fast reversible photographic media is also required to provide a true read/write memory that can be considered a replacement for either tapes or discs. Several such photographic media, which are in different stages of research and development, offer promise for providing the necessary read/write capability. However, in general these media tend to require greater energy or a longer time for writing and erasing than conventional non-reversible silver halide film. This tends to rule out CRT writing and puts even greater importance on the development of suitable lasers.

ON-LINE APPLICATIONS

One of the more important trends in computer applications and systems design is the ever increasing tendency to use computers in an on-line mode of operation. The large time-sharing systems are only one of many examples of this trend toward putting the user directly on-line with the computer. From the peripheral equipment standpoint, increased on-line usage implies the need for large quantities of better user-oriented terminals and increasing requirement for on-line mass storage. The widely used Teletype has many disadvantages as a computer terminal, but it has one major overriding advantage: low cost. Higher speed, quieter, more reliable terminals are required that can sell in the same price range as the Model 33 Teletype. A number of terminals that offer advantages over the conventional teletype are now on the market, but their cost is usually three to four times that of a standard Teletype.

On the other extreme of on-line usage involving man-machine interaction are engineering design and other graphic applications which require sophisticated high-speed CRT terminals with both graphic input and graphic output capability. Terminals of this type are presently priced in the $20,000 to $100,000 price range, although several slower speed graphic terminals using storage tubes are available in the $10,000 range. Alphanumeric CRT terminals priced in the $3,000 to $10,000 price range fall between the Teletype or electric typewriter on the low-cost end and the high-performance graphic CRT terminals on the high-price end. On-line use of computers through terminals of these types combined with inexpensive large capacity mass storage will permit the user to interrogate the computer when he wants a particular answer, thus avoiding the need for printing voluminous reports which are used essentially on an exception basis.

A major need is the development of specialized terminals which are designed for specific types of users. A Teletype, electric typewriter, or an alphanumeric CRT terminal is, in a sense, a general purpose terminal that can be used by many different kinds of users but which is not ideal for most of them. There are a number of different large classes of potential on-line users, such as accountants, retail stores, real estate agencies, etc., which need terminals designed specifically for that type of user. The terminal that is good for one such class of users is not likely to be optimum for a different class.

Finally, on-line usage of computers includes not only man-machine interactive systems but also systems in which instruments, control devices, and other types of physical (and usually analog) equipments are tied directly into the computer. Examples of such systems include automated test and diagnosis systems, data acquisition systems, medical laboratory automation systems, air and water quality monitoring systems, and many others. In a broad sense, the instruments, indicators, valves, and other equipments used in such systems can be considered computer peripheral equipment. At least this is true from the viewpoint of the computer designer, although the instrument man probably would view this quite differently. The important point is that the use of instruments and other physical devices directly on-line with computers will both require and permit significant changes in the design of the instruments themselves. Hence, the next few years should see the development and marketing of a number of instruments and control devices which are designed specifically for use in a computer controlled system. Some of these will, in fact, include a small computer as part of the instrument or control device itself.

IMPORTANCE OF BETTER PERIPHERAL EQUIPMENT TO THE FUTURE OF THE COMPUTER INDUSTRY

It is very likely that during the next decade the growth of the computer industry will be tied to progress in the development of peripheral equipments. Peripheral equipments should and will demand and receive a larger share of the talent and funds expended in the computer industry than has been the case in the past.

SOFTWARE IN THE SEVENTIES

Werner L. Frank, Informatics, Inc.

Software is becoming more expensive, whereas processing is decreasing in cost. A line of code increased in cost by as much as 50 percent during the past decade. In the same period, the cost of performing a data processing calculation was reduced by at least as much. Thus, as we enter the Seventies, there is an increasingly greater divergence in the hardware/software cost composition for EDP. In fact, a 1968 McKinsey & Co. report indicates that, at that time, only 35 percent of the cost of a data processing activity was for equipment expenditures, whereas 65 percent was for operations and programming.

Cost, however, is not even the whole story. Reliability of software delivery schedules, with attendant long lead times, is another factor which is now more readily appreciated, but which is still not tolerated. Thus the market place will respond positively to techniques which will serve to reduce system implementation time periods.

The hardware/software split will certainly cause a response from the EDP industry in order to set these economic forces in better balance. The time and expenditure factors for software development will lead to innovations which will seek to reduce these cost elements. Specifically, we should expect:

1. Development of generalized application packages which can be used by a broad segment of industry (e.g.,payroll package, accounts receivable package).

2. Development of programming aids, languages, and tools to facilitate the programming production task (e.g.,file management systems, automated flowcharting).

3. Embodiment of more software capability in the hardware itself (e.g., hard-wired software or "firmware").

These efforts will, at best, serve only to contain the software cost explosion. There are, however, other factors that are influencing the marketplace which will further enhance software requirements. These factors include:

1. The development of newer hardware (e.g., fourth generation) and accompanying basic software needs. We need only remember, for example, that IBM required some 5000 man-years to develop the operating system for the 360 line of computers.

2. The ever-increasing complexity of applications which are committed to automation. Specific requirements by industry alone account for large programming efforts, such as, for example, the estimated 400 man-years invested by IBM in the development of the Programmed Airlines Reservations System (PARS).

3. The widening technology base of the EDP market itself. The Seventies are heralding the broad use of communications-oriented data-processing systems. In this connection, the proliferation of wide classes of terminal devices and automated data entry will create a new variety of software needs not fulfilled in the Sixties, such as on-line application development (to be carefully distinguished from presently performed on-line programming via time-sharing

systems). The software impact of terminals has not yet really been felt. For example, in 1968 this writer estimated that approximately 100,000 CRT terminals would be on-line in 1972. It now appears that this figure was underestimated by a factor of two or more.

In addition to the new software needs identified above, the attempt to meet the cost reduction objectives cited earlier will also lead to expanding software activity. In fact, these objectives are the basis of the burgeoning "program product" market. Innovative software products are already available which, when utilized, serve to increase programming efficiency by several factors. Such products become supportable entities in their own right and require substantial amounts of investment.

Having thus identified the growing size of the software market itself, it is important to note how that market will be exploited. The software industry can, of course, benefit directly from this expansion, but it will also have to change its character if it is to meet this challenge and reap the resulting benefits.

The hardware manufacturers' unbundling action during 1969 has served to endorse the concept of software for purchase. It also lends credence to the importance and size of this market. This acceptance and participation by the manufacturer in the program product business will ultimately reflect the engineering, marketing, and support approaches which have been successfully employed by these manufacturers in purveying their EDP hardware. Thus, software companies will require a market sophistication which they presently do do not have. A fundamental realization which will occur quickly is the necessity for proper pricing of program products in light of the expected "service costs." Today, this pricing is not fully appreciated and, as a result, recent program products available from independent software companies are significantly underpriced.

Thus, it can be expected that participation in the program product market will require much larger investments than presently envisaged. Accordingly, the program product market will favor larger software companies and, of course, the hardware manufacturers themselves. The smaller entities of the software industry will have the additional basic problem of developing a marketing organization. Because of the tremendous costs involved, it could be expected that either the larger companies will serve this function on their behalf or independent sales representative organizations will emerge to serve this need.

In conclusion, the following estimate of the software market size is presented, giving due consideration to the preceding discussion and excluding the markets of training and service bureau operation:

	Software (Billions of Dollars)		
Software Products	1968	1975	1980
By Software Industry	.01	1.0	3.0
By Hardware Mfr.	0	1.0	2.0
Custom-Developed by Software Industry	.26	1.4	2.5
Free from Mfr.	.23	.4	.5
In-House Developed	3.60	6.2	7.0
Total Software	4.10	10.0	15.0

MEMORIES - TECHNOLOGY AND MARKETS FOR THE 70'S

George R. Cogar, Cogar Corporation

TECHNICAL JUDGEMENTS: BUSINESS DECISIONS

Memories for computers will represent the largest dollar volume for an identifiable product entity in the electronics industry for the 70's. Early in the decade, monolithic semiconductors will become the dominant memory technology representing more than half of the new sales by 1975 and constituting the majority of in-use main memory before the close of the decade.

There are a number of compelling reasons for this to occur - most significantly:

> Semiconductor memories offer raw performance in terms of speed and density which is unachievable in other current technologies. With sub-100-nanosecond memories now in production and new designs having another order of magnitude increase in speed, now in the laboratory, semiconductor memories constitute an exclusive performance class.
>
> Monolithics, being a homogeneous technology, offers the potential of entirely new system architectures. These new architectures will involve parallelism to a degree where today's definition of memory and processor will seem nonsensical. Implementation of memory hierarchy will provide transparency for all levels of programming. Associative memories will become economically feasible and it can be expected that associative concepts will be extended to include synoptics. These new architectures can be expected to contribute substantially toward the solving of the problem of software economics.
>
> Being a process-based technology, monolithics have an intrinsic reliability unachievable by technologies involving a piece part assembly base. Further, where the assembly-based technologies have reached, or are rapidly approaching, a plateau in terms of mechanization and will thus have their manufacturing costs directly tied to labor, monolithics have the potential for significant increases in densities, yields and packaging innovations.

Viewed from a different perspective, one which encompasses overall computer technologies for the 70's and beyond, monolithics must become the dominant technology if for no other reason than simply that the rate of expenditures within the industry has become so high that we cannot now afford a transitional technology having a known obsolescence. While the inertia of capital investment and engineering and manufacturing skills will be a pressure to extend the life of magnetic technology into the 70's, the threat of early competitive impact of systems utilizing magnetics will cause monolithics to prevail.

To this date, to a great extent, the management of our industry could be characterized as a tug of war between the engineer and the marketeer. While this was probably a necessary part of the development period of our industry, it is now a luxury we can no longer afford. We have reached a level of business where we can no longer afford to innovate for the hell of it or to market solely on the basis of a faster, more exotic device.

With the change to principally a process-based technology, one characterized by very high capital investment and a heavy scientific supporting base, many companies must necessarily re-examine their proprietary posture of what and from whom they buy and what and to whom they sell. To those companies who continue to plan their business solely to maximize value added in the production of hardware, great trauma lies in store.

Where the engineer and the marketeer previously made decisions, they will now provide judgement contributing to a more broadly based business decision. Technologies having reached a plateau that is sufficiently broad to discourage the expenditures necessary to attempt the scaling of the next precipice, efforts will now be concentrated on the refinement of technologies and broadening of the market base.

DIGITAL IC MARKET AND TECHNOLOGY TRENDS

Robert B. Seeds, Research and Development Laboratory
Fairchild Camera and Instrument Corporation

In projecting market trends within the semiconductor industry relative to computing applications, it is necessary to first examine the rapid historical progression of digital component technology and then extrapolate its probable effect on the marketplace. The 1960's could be referred to as the decade of digital integrated circuits with their introduction and subsequent development to a point of almost exclusive use in the design and construction of data processing systems. Major achievements during this period related to improvements in circuit performance, broadening of product lines and reductions in volume production costs of the various gate and flip flop "building block" components. Towards the end of this period came the introduction of more complex Medium Scale Integration (MSI) bipolar parts such a 4 bit registers, 2 bit adders, 8 bit multiplexers, etc., and Large Scale Integration (LSI) MOS parts such as 256 bit memories and complete 8 bit arithmetic logic units. It is likely that the 1970's will be most notable in viewing a proliferation in usage of LSI which will certainly continue to impact logic applications but will be even more spectacular in the field of data storage where by 1975 semiconductors should replace cores as the technology for new "mainframe" memory systems.

Solid state random access memory (RAM) components obtainable today include 256 bit modules having 1 to 2 microsecond access/cycle capability at 5 to 10 cents per bit (MOS) and 64 bit elements with 75 nanosecond access/cycle at 15-30 cents per bit (BIPOLAR). By 1973-74 it is expected that MOS type components will allow 250 nanosecond megabyte system performance at a cost of around one cent per bit and BIPOLAR elements will provide megabyte systems with 100 nanosecond performance at a cost of 2-3 cents per bit. RAM requirements may be divided into the following categories: 1) low capacity "buffers" for terminals and peripherals, 2) medium capacity high speed "scratchpad" to enhance the virtual performance of the CPU, 3) "main"store or working program storage and finally 4) "mass" store having very large high speed file capacity of greater than 10^8 bits. Initial penetration of solid state components will be in the first two categories with an estimated 5×10^8 bits being manufactured for sale at an average price of 10 cents per bit in 1970. By 1975 the third area will be under assault as some twenty five billion RAM bits will be fabricated for sale at an average price of 2 cents per bit leading to a total dollar sales volume estimated at 500 million dollars. In fact, major systems manufacturers are expected to announce large data processing configurations utilizing solid state "main" memory by the end of this year or early next year.

Although high complexity logic arrays are also achievable and, in fact, already available within the industry (viz 200-300 gate custom MOS) widespread usage in large central processor applications will be limited due to the logistics difficulties of functional standardization, i.e., "the parts number problem." Using simple Boolean building blocks such as gates and flip flops, the number of part types required for efficient logic design is on the order of 10. With MSI utilization where logic structure is regular (such as registers, arithmetic processing, and data routing) some 30 to 50 per cent of the gates in a CPU can be effectively implemented by adding only about 10 more part types. However, it is estimated that to partition a central processor system of present day architecture into 100-200 gate functions would require 100 to 200 part types, the bulk of which would be used only once in the system. New system architectures allowing LSI standardization, particularly in respect to regularity of control structure, will overcome these problems in time but the major application of logic LSI in the early seventies will be in airborn computers, small special purpose processors, "mini" computers, terminals with processing capability and other configurations where the production of a large volume of equipment can amortize the custom development charges. These applications are not to be minimized however and large development programs exist within the semiconductor industry aimed at custom logic array process fabrication technology and design technology via computer aided systems. By 1975, 250 gate functions will cost roughly five thousand dollars to have developed and sell for 3-4 cents per gate in 1000 up quantities. Even excluding the large central processor it is estimated that the total dollar market for logic LSI may well keep pace with that for memory and both will be limited by the semiconductor industry's ability to build capacity more than any other single factor.

These trends in product capability will have certain predictable impacts on our industry. First of all, the technical interface between the component vendor and the customer will rise another level from that of a circuits orientation to one of a logic and systems orientation, thus the ability to plan and develop new products and to communicate with the customer at this level will in large measure determine a given vendor's success. Secondly, the availability of low cost electronics will create broad new markets for terminals and memories which will guarantee a continued rapid growth of the total dollar volume of digital components. Thirdly, with the system's expertise which is evolving within the semiconductor industry, and with the value added relative to the above mentioned market areas for LSI being principally electronic component oriented, a most interesting trend will be the emergence of the integrated circuit manufacturer as major subcontractor and or supplier of "black box" sub system hardware.

SEMICONDUCTOR CHIP INTERFACING

T. J. Matcovich, Drexel University, Philadelphia, Pa.

ABSTRACT

The problems involved in making connections to monolithic circuit chips become more difficult as the number of leads per chip becomes greater and the chip pad size and spacing becomes smaller. Several of the methods devised to make interconnections from chips to external wiring are described in this paper. The methods include flying-lead, beam-lead chip, solder-ball and beam-lead substrate techniques.

Some of the metallurgical problems encountered with the various techniques are described, and manufacturing problems are discussed. Guidelines are presented for the choice of an interconnection technique.

INTRODUCTION

An important decision in the use of integrated circuits is the choice of method for making electrical connections to the circuit chips. The flying-lead was the first technique used for this purpose since the technology was available from the preceding work on transistor packaging. As the integrated circuits became more complicated, more connections had to be made to each chip and more effective interconnection techniques were developed.

As the complexity of integrated circuits continues to increase, the use of one of the new interconnection techniques becomes essential. Consider the data presented in Table I. These data indicate the dimensional limitations imposed by the layout of the integrated circuit; the extremes indicated cover the range of what is presently available to what might be expected within a few years.

TABLE I

TYPICAL INTEGRATED CIRCUIT DIMENSIONS

	Minimum	Maximum
Chip Size	50 x 50 mils	200 x 200 mils
Thickness	2 mils	12 mils
No. of Pads	10	100
Pad Size	2 x 2 mils	5 x 5 mils
Pad Spacing	4 mils	20 mils

Ten connections to a 50 x 50 mil chip can easily be made by using flying-leads although other types of interconnections may be less expensive and more reliable. One hundred connections on 4-mil centers cannot be made with reasonable yield in a production environment by using flying-leads, and some other technique must be employed.

Several techniques which have been developed for making interconnections are described in this paper. The limitations of the techniques are described and guidelines for the selection of an interconnection technique are suggested.

DESCRIPTION OF TECHNIQUES

The numerous techniques developed for making electrical connections to integrated circuit chips can be divided into four groups. The original and still most popular technique is the flying-lead which was borrowed directly from transistor technology. This interconnection is shown schematically in Figure 1. The flying-lead is typically one mil in diameter and made from gold or aluminum. Aluminum wires have a few per cent of silicon added to prevent the migration of silicon from the chip into the aluminum wire. The wires are bonded at the chip end by thermocompression bonding or ultrasonic bonding. In thermocompression bonding the parts to be joined are clamped together and heated. The temperature is maintained below the melting point but is sufficient to soften the materials. Under the influence of the clamping force the members to be joined are deformed slightly breaking oxide and other contaminant layers and bringing the two members into intimate contact. The contact is so close that the members are bound by molecular forces. Ultrasonic bonding results in same kind of bond; however, the members to be joined are not intentionally heated. The members are clamped together and ultrasonic energy is applied so as to produce a scrubbing action at the interface. This scrubbing action destroys oxide and other contaminant films and the members are brought into intimate contact under the influence of the clamping force. There are conflicting data as to whether the temperature at the interface rises significantly[1,2]; however, since no thermal energy is deliberately added in the bonding process it may be classified as a low temperature bond.

If aluminum wire is to be bonded to aluminum chip metallization ultrasonic bonding must be used. Good bonds are sufficiently strong that the wires fail before the bond.

The second group of techniques is referred to as flip-chip or direct bonding. This interconnection is shown schematically in Figure 2. A pedestal or stand-off is formed at each pad location and the chip is connected to the substrate wiring via the pedestals. The pedestals may be formed on the chip or on the substrate and may be made from a number of materials. The bond is made by thermocompression bonding, ultrasonic bonding, or solder bonding.

This group can be logically divided into those techniques employing solder and those not employing solder. The solder technique is exemplified by the IBM approach. For the IBM SLT modules a copper ball surrounded by solder is used for a pedestal.[3] This approach is practical for 3 pad chips, but not for chips with significantly larger number of pads. For the more complicated chips only solder is used for the pedestal.

The metallurgy used for the SLT units comprises several levels of metals. Aluminum is used to provide ohmic contact to the silicon device; chromium is used as an adhesive layer and also as a diffusion barrier between the solder and the aluminum; copper provides a solderable layer above the chromium; a phased copper-chromium layer is formed between the chromium and copper layers and is used to prevent the solder from reaching the chromium film; a lead-tin solder is formed on the copper layer and a copper ball is placed on top. The copper ball is nickle plated to prevent the formation of copper-tin intermetallic compounds on the ball. When the solder is reflowed the ball is surrounded by the solder. The ball provides the stand-off or pedestal. No ball is used for the more complicated chips[4]; the solder provides the stand-off or pedestal. For these chips care is taken to limit the solder-wettable area on both chip and substrate. When the chip is bonded, surface tension holds the solder ball in shape and prevents its complete collapse. Solder bumps are usually formed on the wafer before the chips are separated.

The non-solder techniques vary widely. A common pedestal material is aluminum and the pedestal may be on the substrate[5] or on the chip.[6] Copper is sometimes used and bonding directly to thick film materials has been reported.[7] Bonding is accomplished by thermocompression or ultrasonic bonding. Complicated metallization schemes are not required. Thin films are used on the substrate and the wiring frequently includes a layer of chrome, molly or nichrome to act as an adhesive layer. Shear bond strengths as high as 80 grams per pad have been achieved, and techniques for removing and replacing defective chips have been developed.[5] Systems employing aluminum must be protected since aluminum corrodes rapidly in warm, humid ambiants.

The third group of techniques is referred to as beam-leaded chips. This interconnection is shown schematically in Figure 3. Small leads are formed on the chips prior to wafer separation. These leads extend beyond the edge of the chip where they can be aligned and bonded. The beams are made of gold or aluminum and are usually thermocompression bonded. The gold beam lead is exemplified by the Bell Laboratories' approach.[8] The metallurgy used comprises several layers of metals as shown in Figure 4. Platinum silicide is used to make ohmic contact to the silicon. This is followed by a thin titanium layer which serves as an adhesive layer to the silicon nitride. The titanium is coated with platinum and then gold. A thin layer of gold is used to form the interconnections on the chip and a thick layer to form the beam leads. The platinum acts as a barrier to keep the gold out of the silicon and to prevent its interaction with the titanium. The chips are normally passivated by depositing a layer of silicon nitride and a coating of silicone resin. The latter material minimizes the surface migration of the gold and the former material protects the silicon surface from moisture and other contaminants. Since the beams extend beyond the edge of the chip, the chips cannot be separated by a scribing or cutting process. To separate the chips, the wafers are reduced in thickness by lapping, and then etched from the back. Methods have been devised to hold the chips in the matrix array after separation for ease in subsequent handling.

Several companies have done development work on aluminum beam leads. The use of aluminum for the beams results in a simpler metallurgy and simpler fabrication processes. However, systems made from aluminum must be protected because of the rapid corrosion of aluminum in warm, humid ambiants. At least one company[9] is using aluminum beam leads in a commercial system.

The fourth group of techniques include beam-leaded substrates[10,11] and some related special techniques such as GE's STD process[12] and Motorola's "spider".[13] The beam-leaded substrate is shown schematically in Figure 5.

The beam-leaded substrate is exemplified by Motorola's approach to packaging a small memory.[11] A printed circuit board of 1-mil-thick Kapton dielectric film has conductor patterns formed on both sides and interconnected by plated-through holes. In areas where integrated circuits are to be attached a finger pattern is formed on one side of the film. The dielectric film is etched from the back of the finger area and the fingers are individually bonded to the pads on the chip. The chip is bonded to a ceramic base which also contains a layer of conductors. The ceramic provides a heat sink for the chip and a support for the printed circuit board. The beam-lead material is nickel and the chip metallization is aluminum; an ultrasonic bonding technique is used to join the nickel to the aluminum.

The General Electric STD technique[12] is shown in Figure 6. A lower level of conductors is formed on a substrate. Mesas are formed where feedthroughs are required to the second level of conductors and a plastic layer is bonded to the substrate by a thermocompression process. The plastic layer is of the same height as the mesas, and the tops of the mesas are left exposed. Active components are positioned over the plastic and the assembly is heated until the plastic becomes fluid. The chips settle into the liquid plastic and on cooling are firmly held in place. The top surface of the chip stands exposed and a second level of conductors is evaporated on the surface to complete all interconnections. Note that all discretelead wiring operations have been eliminated.

In contrast to the exotic STD process the Motorola "spider" technique is intended for use as a cost-reducing production tool. Aluminum lead frames formed in continuous rolls of material are bonded to the pads on the chips. The aluminum lead frame is then welded to a more substantial lead frame which will eventually form the external leads. The excess aluminum is removed and the chip is plastic encapsulated while still on the continuous roll of lead frames. Two bonds must be made for each lead as in the flying lead technique; however, all of the chip bonds are made simultaneously as are all of the bonds to the external lead frame. Since both sets of lead frames come in continuous rolls, once the chip is bonded the handling of the package is relatively simple and considerable automation is possible.

Another interconnection approach is the leadless inverted device (LID). The LID is actually an intermediate package and does not fall naturally into the specified groups. It is a ceramic structure which has a recess for mounting the chip and a series of electrically isolated conducting legs. The chips are wire bonded to the legs within the cavity and external connections are made by soldering to the legs. Nothing is gained over the conventional flying-lead package except size and perhaps cost. Consequently, this technique as well as the spider technique represents transitional type interconnection schemes which will have limited long range application, but perhaps significant short range utility.

METALLURGICAL CONSIDERATIONS

The interconnection techniques described range from simple all aluminum systems to complicated multilayer structures. A large number of metallurgical problems are experienced in all of these systems.

The simple all aluminum systems might be expected to be trouble free; aluminum makes good ohmic contact to appropriately doped silicon and adheres well to the oxide layers on both chips and substrates. However, silicon diffuses into the aluminum and this can result in device failure,[14] aluminum migrates under the influence of high current densities and this can cause failure,[15] and aluminum corrodes rapidly in warm, humid ambiants[16] and this can cause failure.

The solder systems are complicated metallurgical systems. Materials which are used to make ohmic contacts to silicon generally cannot be wet by solder; layers of additional materials must be added as barriers to prevent contamination of the silicon by the solder materials and as adhesives to improve adhesion of subsequent layers. The problems experienced with the IBM solder technique are described in the literature.[17]

The beam-lead chip and beam-lead substrate systems have similar problems of adhesion, contamination and interaction. For example in Bell's beam-lead system the gold must be isolated from the titanium since they will interact to form undesirable intermetallic compounds. Adhesion to the dielectric is a problem in many of the beam-lead substrate systems.

In general, materials must be chosen to survive the affects of subsequent processing steps as well as the ultimate environment. The early experience with failures due to intermetallic compound formation in gold-aluminum-silicon systems has made all multi-metal systems suspect. Since metallurgical degradation is a process which can continue while the system is in use, reliability testing is a time consuming operation. Excellerated life tests must be interpreted carefully since the importance of some failure modes may be exaggerated[18] under the test conditions.

Significant quantities of reliability data exist only for the flying-lead technique, Bell's beam-lead technique, and IBM's solder technique.

MANUFACTURING CONSIDERATIONS

Special Preparation

A minimal amount of special preparation is required when flying-leads are used. All other techniques require special preparation of the chips or the substrate or both. Those which require special preparation of the chip include beam-lead techniques, solder techniques and some variations of non-solder, direct-bonding techniques. There are several advantages to doing the special preparation on the chip level: the chips can be batch processed while still in the wafer form, the circuits can be passivated at the chip level, and the circuit may be tested more conveniently than in other approaches. The disadvantage of these approaches to users who do not have the required technology or facilities is obvious. These users must depend on the availability of suitably processed chips from suppliers. If large quantities of chips are required this problem will be minimized, but if small quantities are required it may be insurmountable. Other disadvantages include: The cost of the chips is increased, special handling facilities may be required, and special bonding equipment may be needed.

Those techniques which require special preparation of the substrate include the beam-lead substrate, the STD process and some variations of the non-solder, direct-bonding techniques. There are several advantages to doing the special preparation on the substrate level: most integrated circuit chips can be used in standard form, the special preparation is done on a low priced substrate rather than a high priced wafer, and the technique is compatible with the use of a large variety of chips and components. The disadvantages include: increase cost of the substrate, loss of passivation at the chip level, and lack of reliability data.

Equipment

Ultrasonic and thermocompression wire bonding equipment is generally available and of good quality. Equipment for direct bonding is also available. If the direct bond is to made to an opaque substrate (the usual case), some means must be provided to align the chip pads or pedestals to the substrate pedestals or pads. Three approaches are presently used: a half silvered mirror is placed between substrate and chip so that both may be viewed simultaneously and positioned for coincidence, or the chip and substrate are separately aligned to the same crosshair in the optics and are then mechanically transferred to an aligned position, or the chip is aligned automatically by using the chip edges as references.

Equipment is also available for beam-lead bonding. Three techniques are used; all are based on thermocompression bonding. In one case a picture frame shaped tool is brought into simultaneously contact with all the leads. In the second case a tool of similar shape is caused to roll or wobble[19] around the chip edge thereby contacting only a few of the leads at one time, but completing all bonds in one machine operation. In the third case, called compliant bonding,[20] an aluminum picture frame is placed around the chip, but in contact with the leads, and a flat faced tool is brought into contact with the picture frame to apply the bonding pressure.

Equipment for solder bonding is also available. In addition to furnaces needed for reflow, equipment is also needed to place the chip in position.

In general bonding equipment for any of the interconnection techniques is available and the industry is turning its attention to automated handling and testing to increase speed of operation.

Packaging

The choice of interconnect technique depends on the type of package used and the end use of the product. In general the integrated circuit must be protected from its environment. Beam-lead and solder techniques allow for the circuit to be passivated at the chip level. Other techniques require sealing at the package level especially if there is aluminum in the system. Plastic encapsulation is a simple and inexpensive method for sealing circuits; however, there is still some question about its effectiveness in some environments. Hermetically sealed cases are frequently used for isolating the circuits from the environment, but such packages tend to be awkward when used with large hybrid circuits and their use precludes system repair once the seal is made.

Although passivating at the chip level is most desirable and therefore suggests the use of beam-lead or solder techniques, other considerations may be more important in the choice of interconnect technique.

CHOICE OF TECHNIQUE

A number of alternatives to the use of flying-leads for making connection to integrated circuits have been developed to the extent that their use should be seriously considered in new systems. The choice of a technique must be an engineering decision based on economic as well as technical feasibility. The first decision to be made is: "Which of the techniques is technically feasible for the proposed system?" The technical limitations of the various interconnect techniques are shown in Table II. The data in the table are intended to offer only a rough guideline. For example, if the unit will be subject to high temperatures in subsequent packaging operations or in final use, then solder bonding may not be adequate.

The second decision to be made is: "Can the process be developed in the available time and is the cost reasonable?" Data intended as a rough guideline to answer these questions is presented in Table III. Clearly the answers will depend on the individual circumstances. Bell Laboratories, IBM, Government Contractors, Integrated Circuit Houses and Systems Houses could each consider the same problem and arrive at different but logically correct choices for interconnect technique.

TRENDS

Flying-leads are still widely used and may represent the best approach to a current problem; however, definite trends towards the use of other techniques are developing. The techniques most likely to become dominant are beam-leaded chips and solder interconnections. A major disadvantage of these techniques is the limited availability of prepared chips. In the case of beam leads this condition is changing. Motorola, Texas Instrument, Raytheon, Sylvania and General Instrument are all either offering or planning to offer gold beam-leaded silicon devices. If an appreciable variety of beam-leaded devices become available, this interconnection technique will become dominant.

TABLE II

TECHNICAL LIMITATIONS OF INTERCONNECTION TECHNIQUES

	Passivated at Chip Level	High Temperature Operation	Compatible with			Reliability	Thermal Dissipation
			Thick Film	Thin Film	High Lead Density		
Flying-Lead	No	Yes	Yes	Yes	Poor	Good	Good
Direct Bond Solder	Yes	No	Yes	?	Good	Good	?
Direct Bond Non-Solder	No	Yes	?	Yes	Excellent	Insufficient Data	Fair
Beam-Lead Substrate	No	Yes	?	Yes	Good	Insufficient Data	Good
Beam-Lead Chip	Yes	Yes	?	Yes	Excellent	Good	Poor

TABLE III

ECONOMIC LIMITATIONS OF INTERCONNECTION TECHNIQUES

	Chips		Bonding Cost per Chip	Technology		Suitable for Small Quantity Production
	Availability	Cost		Developed	Available	
Flying-Lead	Excellent	Low	High	Yes	Yes	Yes
Direct Bond Solder	Poor	High	Low	Yes	?	No
Direct Bond Non-Solder	Excellent	Low	Low	Yes	Yes	Yes
Beam Lead Substrate	Excellent	Low	Low	?	?	Yes
Beam Lead Chip	Poor to Fair	High	Low	Yes	?	No

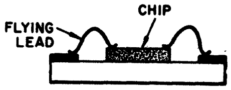

CHIP
FLYING LEAD

Fig. 1 FLYING-LEADS

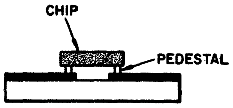

CHIP
PEDESTAL

Fig. 2 DIRECT BONDING

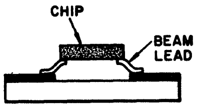

CHIP
BEAM LEAD

Fig. 3 BEAM-LEADED CHIPS

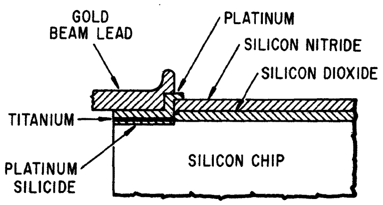

GOLD BEAM LEAD
PLATINUM
SILICON NITRIDE
SILICON DIOXIDE
TITANIUM
PLATINUM SILICIDE
SILICON CHIP

Fig. 4 BEAM-LEAD CROSS SECTION

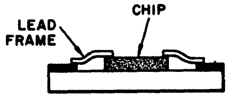

LEAD FRAME
CHIP

Fig. 5 BEAM-LEADED SUBSTRATES

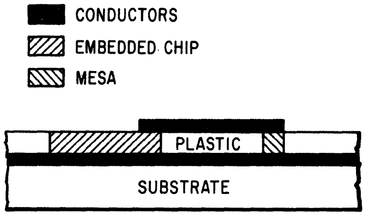

CONDUCTORS
EMBEDDED CHIP
MESA
PLASTIC
SUBSTRATE

Fig. 6 STD TECHNIQUE

REFERENCES

1. Peterson, J.M., "Ultrasonic Welding in Electronic Devices," IRE International Convention Record (1962).
2. Ohara, H., Sato, K. and Kamoshita, G., "Metallurgical Study of Flip-Chip Bonding." To be published. Central Research Laboratory, Hitachi Kokubunji, Tokyo, Japan (1969).
3. Totta, P.A. and Sopher, R.P., "SLT Device Metallurgy and its Monolithic Extention," IBM J. Res. Develop. 13, 226-238 (1969).
4. Miller, L.F., "Controlled Collapse Reflow Chip Joining," IBM J. Res. Develop. 13, 239-250 (1969).
5. Matcovich, T.J., "Memory Systems for Microcircuits (Manufacturing Methods)," Technical Documentary Report No. AFML-TDR-66-42 (1966).
6. Rodgers, B. and Reynolds, J.A., "How Fairchild Makes its New Facedown Bonded Microcircuits," Insulation, 48-51 (November 1967).
7. Matcovich, T.J., Coren, R.L., Kelleman, D.G. and Galli, R.J., "Ultrasonic Bonding of Integrated Circuits to Thick Film Pedestals on Ceramic Substrates," Proceedings Electronic Components Conference, 254-275 (1969).
8. Hause, S.S. and Whitner, R.A., "Manufacturing Beam-Lead Sealed-Junction Monolithic Integrated Circuits," The Western Electric Engineer XI, 2-15 (1967).
9. Computer Microtechnology, Inc., 610 Astoria Street, Sunnyvale, Calif. 94086.
10. Bachner, F.J.,Cohen, R.A. and McMahon, R.E., "Rigid and Nonrigid Beam-Lead Substrates," Solid State Conference Digest of Technical Papers, 94-95 (1970).
11. Marley, J. and Trolsen, G., "New Beam-Lead Connection Method Boosts Semiconductor Memory Yields," Electronics 42, 105-110 (1969).
12. Suran, J.J., Kim, C.S. and Palmer, G.G., "Plastic-Structured Microelectronics," Solid State Conference Digest of Technical Papers 96-97 (1968).
13. Anon., "Spider Bonding Technique," Electro-Technology, 35 (March 1969).
14. Schnable, G.L. and Keen, R.S., "Aluminum Metallization - Advantages and Limitations for Integrated Circuit Applications," Proceedings of the IEEE 57, 1570-1580 (1969).
15. Black, J.R., "Electromigration Failure Modes in Aluminum Metallization for Semiconductor Devices," Proceedings of the IEEE 57, 1587-1594 (1969).
16. Moore, R.P., "Reliability Test Program of Ultrasonic Face Down Bonding Technique," Technical Report No. RADC-TR-138 (June 1967).
17. Berry, B.S. and Ames, I., "Studies of the SLT Chip Terminal Metallurgy," IBM J. Res. Develop. 13, 286-296 (1969).
18. Schnable, G.L. and Keen, R.S., "Metallization and Bonds - A Review of Failure Mechanisms," Sixth Annual Reliability Physics Symposium Proceedings, 170-192 (1967).
19. Bycer, M., "Principles of Beam Lead Bonding," Kulicke and Soffa Industries, Inc., 135 Commerce Drive, Fort Washington, Pa. (1969).
20. Coucoulas, A. and Cranston, B.H., "Compliant Bonding - A New Technique for Joining Microelectronic Components," IEEE Transactions on Electron Devices ED-15, 664-673 (1968).

CERAMICS FOR PACKAGING

D. L. Wilcox, IBM Components Division, East Fishkill Laboratory, Hopewell Junction, N.Y. 12533

ABSTRACT

Ceramics are playing a vital role in the solution of the advancing electronic packaging needs by virtue of:

1. the broad range of properties they possess and
2. the product design flexibility due to ceramic processing versatility.

These features have combined to make it possible for the electronic semiconductor package to take on increased functionality allowing increased exploitation of the cost, performance and reliability advances of the semiconductor industry. With the increased functionality comes, however, a new set of problems (part number explosion, testing, turn-around time) for which solutions must be found.

The significant properties of ceramic packaging materials are presented and some of the fundamental parameters controlling the critical, physical and electrical properties are highlighted. Important fabrication techniques for preparing simple substrates and more complex multilevel wiring structures are reviewed and their relative merits indentified in relation to the defined packaging needs.

I. THE NEED

There are many ways of describing the challenges being presented to the technology of packaging electronic components. Packaging for computer applications can be broadly defined as the internal intraconnection of active and passive components into an ensemble designed to perform a useful function for a competitive cost, performance and reliability objective. Within the framework of this definition, it can be seen that semiconductor technologies are fulfilling more and more of the packaging needs right on the silicon wafer, and this trend, the creating of MSI and LSI capabilities, will continue. However, subsequent packaging materials and processes must -- in order to capitalize on the advances in cost, performance and reliability gains at the chip level -- respond to more demanding needs as summarized in Table I.

TABLE I. Packaging Technology Needs/Trends.

1. Signal wiring: more levels, finer resolution, better transmission characteristics
2. Finer via grid: close gap between current silicon technologies and existing packaging technologies
3. Power distribution: higher current capacities
4. Thermal dissipation
5. Chip support: single and multiple chip, larger chips
6. Chip interconnection: increased terminals per chip with decreasing spacing
7. Decoupling
8. Environmental protection

Another way of sizing the packaging challenge is through an analysis of the interconnection or wiring grid available for multilevel wiring, remembering the fundamental description of packaging to be that of interconnecting circuit components. Figure 1 shows a hypothetical interconnecting array requiring several levels of packaging. Maximum packaging density is achieved when the intraconnection and interconnection technologies permit the device density at the chip level to be preserved.

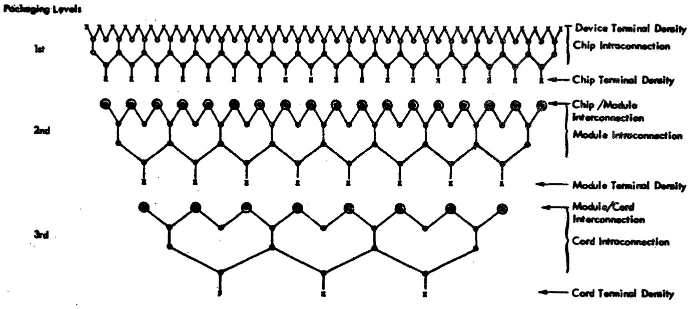
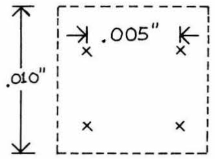


Fig. 1. Packaging (interconnection) schematic.

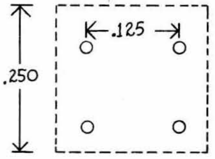
The semiconductor device technoligies are capable, through photolithographic process, of achieving chip terminal densities which far surpass the ability of today's follow-on packaging levels to respond. The gap can be seen by a rough sizing of the via grid capabilities of the semiconductor technology versus the common epoxy glass multilayer printed circuit board.

If one assumes 4 chip terminals per circuit (bipolar technology) and assumes a semiconductor process capability of placing these terminals on a .005" grid (see Fig. 1), a chip terminal density is calculated as follows:



$$\frac{4\ \text{terminals}}{(.01)^2} = 4\times10^4\ \frac{\text{terminals}}{\text{in}^2}$$

IBM's SLT multilayer printed circuit card has an interconnection grid density of:



$$\frac{4\ \text{interconnection vias}}{(.250)^2} = 64\ \frac{\text{terminals}}{\text{in}^2}$$

The challenge to module packaging technologies is to bridge this gap.

A graphic illustration of the impact of the grid gap can be seen in Fig. 2. Figure 2 shows the attendent "fan-out" or space transformation which can occur, and also illustrates the increased density achieved by increasing the connection-density of the module.

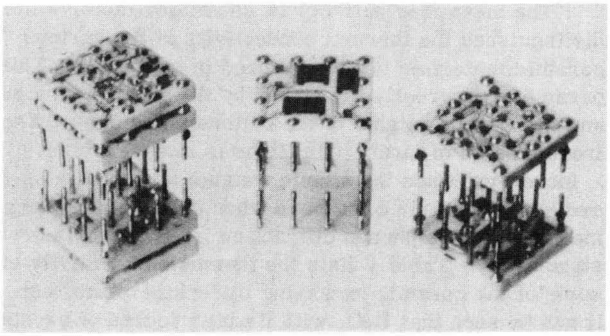


Fig. 2. Comparison of ALST (left), SLT (center) and SLD (right); bottom views are illustrated by mirror reflections.

Ceramics are playing a vital role in the solution of this problem by virtue of the broad range of properties they possess and the intra and interconnection capabilities created by their process flexibility. Significant properties and processes contributing to this capability are reviewed in the subsequent sections.

II. CERAMIC PROPERTIES OF IMPORTANCE

Electrical

The packaging material must provide electrical isolations for the signal wiring networks and also provide a suitable signal transmission environment. The electrical resistivity and dielectric constant of many ceramic materials are quite attractive in this regard. Table II lists these properties for some of the more attractive packaging materials.

TABLE II. Ceramic Packaging Materials: Electrical Properties.

	Resistivity (Ω-cm) 25°C	Resistivity (Ω-cm) 300°C	Dielectric Constant IMHz
Al_2O_3	10^{14}	10^{14}	9.6
BeO	10^{14}	10^{14}	6.5
$3Al_2O_3 \cdot 2SiO_2$	10^{14}	---	6.0
$ZrSiO_4$	10^{14}	10^{11}	8.7
$MgAl_2O_4$	10^{12}	---	8.5
Borosilicate Glass	10^{14}	10^{10}	4.1
Silica Glass	10^{14}	10^{9}	3.8

The conductivity of a solid may be described by the relationship:

$$\sigma = \sigma_i + \sigma_e \qquad (1)$$

where

$$\sigma_i = \sigma_{\text{cations}} + \sigma_{\text{anions}}$$

$$\sigma_e = \sigma_{\text{electrons}} + \sigma_{\text{holes}}$$

and

$$\sigma = nq\mu \qquad (2)$$

where

n = number of charge carriers
e = charge per carrier
μ = mobility of carriers

The ionic contribution to the conductivity of ceramic materials selected for packaging is very small due to the low mobility of the cations and anions selected. Likewise, due to the strong ionic bonding, the electron carrier density is extremely small. Conduction is promoted by: small cations or anions, low charge densities per ion and multi-valence state ions.

Thus, in the selection and formulation of ceramic packaging materials, the trend has been toward high purity oxides with a minimum concentration of small ionic impurities such as Na^+, Li^+, K^+ and multivalence cations of Fe, Ti, Ni, etc.

It is desirable to have the dielectric constant of ceramic materials used in the signal transmission function of the package as low as possible. This, of course, will minimize the signal transmissions delays as:

$$\frac{t_D}{l} = \frac{E_r}{c} \qquad (3)$$

where

t_D = delay
E_r = dielectric constant
l = length of signal path
c = speed of light

Table III lists the propagation delay in nanoseconds (10^{-9} sec) per inch for the selected ceramic packaging materials.

TABLE III. Signal Transmission and Dielectric Constant.

	E_r	t_D/l $\frac{10^{-9}\text{ secs}}{\text{inch}}$
Al_2O_3	9.6	0.260
$ZrSiO_4$	8.7	0.250
$MgAl_2O_4$	8.5	0.246
BeO	6.5	0.230
$3Al_2O_3 \cdot 2SiO_2$	6.0	0.207
Borosilicate Glass	4.1	0.170
Silica Glass	3.8	0.165

The dielectric constant of a material may be expressed as:

$$E_r = \frac{P}{E}$$

where

E = electric field
P = $P_s + P_e + P_i + P_o$
P_s = space charge polarization
P_e = electron polarization
P_i = ion polarization
P_o = dipole polarization

Ceramic packaging materials requiring a low E_r are selected such that P_e and P_i are the only significant contributions to E_r. Table I V lists the polarizabilities of several ions,[1] and it is obvious why low dielectric constant packaging materials are selected to contain Si^{+4}, O^{-2}, Al^{+3} and Be^{+2}.

TABLE IV. Electronic Polorizabilities of Ions.

O^{-2}	3.88	F^-	1.04	Li^+	.029	Be^{+2}	.008	B^{+3}	.003	C^{+4}	.0013
S^{-2}	10.2	Cl^-	3.66	Na^+	.179	Mg^{+2}	.094	Al^{+3}	.052	Si^{+4}	.0165
Se^{-2}	10.5	Br^-	4.77	K^+	.83	Ca^{+2}	.47	Sc^{+3}	.286	Ti^{+4}	.185
Te^{-2}	14.0	I^-	7.10	Rb^+	1.40	Sr^{+2}	.86	Y^{+3}	.55	Zr^{+4}	.37
				Cs^+	2.42	Ba^{+2}	1.55	La^{+3}	1.04	Ce^{+4}	.73

The electron polarizability is related to the electron density (atomic number). The ionic contribution is crystal structure sensitive and related to the strengths of the ionic bond. Thus, ceramic packaging materials requiring low dielectric constants are formulated from ionic specis with low electronic polarizabilities (low atomic number) and from systems containing strong ionic bonding minimizing the ionic polarizability. It is worth mentioning that ceramic materials with extremely high dielectric constants are also possible. Ceramics with E_r from 80 to 10,000 are used as capacitors and are finding increased use in voltage distribution applications.

Thermal

Thermal conductivity and thermal expansion are two of the more significant thermal properties considered in the selection of ceramic materials for electronics packaging. Thermal dissipation considerations are generally given their ample consideration in package design. However, stresses due to thermal properties and mismatch between materials is often treated lightly and can lead to disastrous results.

The thermal and environmental stability of ceramics are also prime factors in their versatility.

The thermal conductivity of ceramic materials in the temperature range of interest (0-400°C) is controlled by the propagation of phonons (lattice vibrations) through the solid according to the relationship:

$$K = sc\lambda/3 \qquad (5)$$

where

s = volumetric specific heat
c = velocity of sound
λ = mean free path of phonon

The mean free path (λ) is the major factor which distinguishes the thermal conductivity of the various ceramic materials commonly used in packaging. This parameter is greatly influenced by the crystal structure and the atomic weights of the cations and anions. The transmission of lattice vibrations is more efficient (i.e., λ increases) when the atomic weights of the ionic specis are low and nearly equal, and when the lattice arrangement minimizes phonon diffraction (i.e., simple crystal strucutres). Table V lists the thermal conductivity of some of the ceramic packaging materials of interest. It can be seen that BeO, with its high degree of crystal symmetry and similarity in atomic weight of its ionic species is an outstanding thermal conductor. Silica glass, at the other extreme, due to its completely random and disordered structure, is a very poor conductor.

TABLE V. Ceramic Packaging Material: Thermal Conductivities.

	K (25°C) (cal-cm/sec-cm²-°C	Crystal Structure	Cation Atomic Weight
BeO	.55	hexagonal-simple	9
MgO	.12	cubic-simple	24
Al_2O_3	.088	hexagonal-more complex	27
$MgAl_2O_4$	.035	cubic-complex	--
$ZrSiO_4$	.018	tetragonal-very complex	--
$3Al_2O_3 \cdot 2SiO_2$	.016	orthorhombic-very complex	--
Borosilicate Glass	.002	random	--
Silica Glass	.003	random	28

In addition to the crystal structure factors which determine the intrinsic thermal conductivity of the ceramic phase, most ceramic materials are utilized as polycrystalline structures and such factors as grain shape, grain orientation, porosity and secondary phases will influence the bulk thermal conductivity.

As shown in Fig. 3, increasing temperature has an adverse effect on K as the mean free path of the phonon is decreased due to the more intense thermal vibrations which create a less regular lattice structure. In the case of SiO_2, the apparent anomaly is explained by the fact that the increase in heat capacity of silica is more significant than any decrease in the λ, which is already very short.

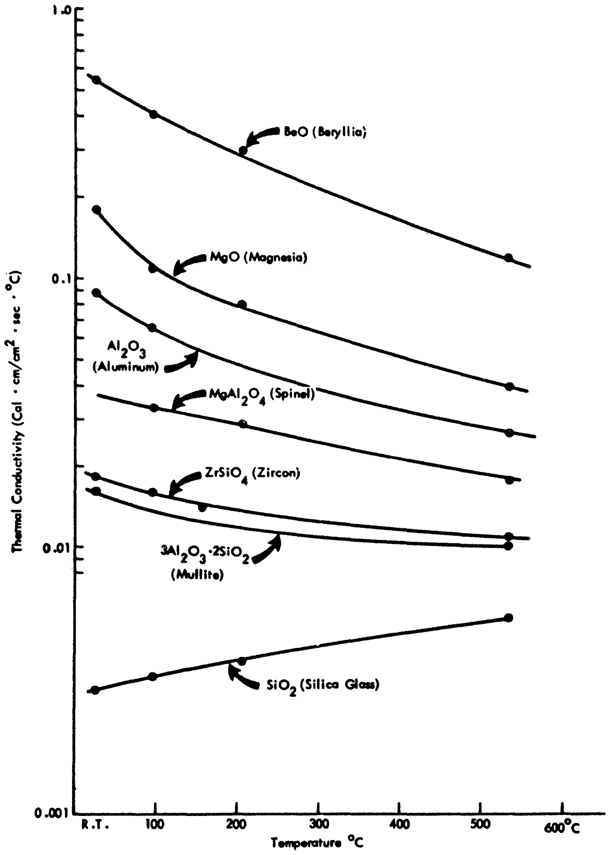


Fig. 3. Thermal transfer (data from J. F. Lynch, et al., "Engineering Properties of Ceramics", AD803-765, U.S. Dept. of Commerce).

The thermal expansion characteristics of ceramics can be explained from the energy diagram shown in Fig. 4 which shows schematically the free energy versus interatomic spacing for a weakly bonded structure and a strongly bonded structure. As shown in Fig. 4, structures which are strongly bonded would be expected to show lower thermal expansion and, inasmuch as melting points of materials are related to the thermal energies required to overcome the attractive forces present, correlation should exist between the thermal expansion of the melting point of these materials. This correlation is evident in Table VI.

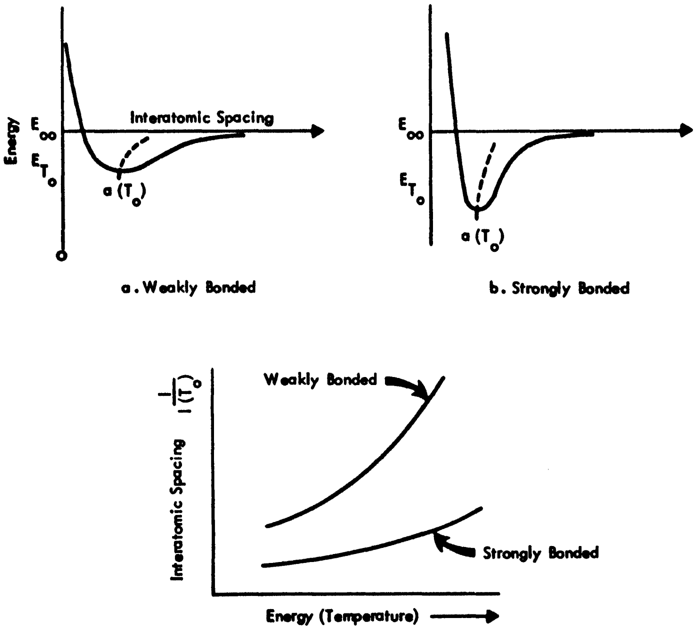


Fig. 4. Bonding energies and thermal expansion.

TABLE VI. Thermal Expansion and Melting Points.

Material	Melting Point (°C)	Expansion (α) at R.T. (x 10^6 in/in/°C)
H_2O	0	90.0
Epoxy	150[b]	45.0
MgO (Magnesia)	2852	10.0
BeO (Beryllia)	2565	8.0
$MgAl_2O_4$ (Spinel)	2135	8.8
Al_2O_3 (Alumina)	2072	7.3
$ZrSiO_4$ (Zircon)	1775[a]	4.0
$3\,Al_2O_3 \cdot 2SiO_2$ (Mullite)	1840[a]	5.3
SiO_2 (Fused Silica)	1500[b]	.55
Borosilicate Glass	780°C	3.3

---

[a] Incongruent melting

[b] Softening temperature

The correlation is not perfect due to other factors related to the crystal structure which are not taken into consideration be evaluation of the energy diagram alone. In general, the expansion of the ceramic materials are low and several materials closely approximate that Si (α = 3.5 x 10^{-6} in/in/°C), making it possible to mount chips directly to the ceramic.

Because of the strong ionic bonds common to the ceramic packaging materials, they are very stable in a variety of ambients and at elevated temperatures. This has made it possible to process these materials in a variety of ambients (atmospheres) which are often necessary in the creation of an electronic package. For example, the need for strong ceramic-metal bonds may dictate the use of reducing (H_2) atmosphere at temperatures as high as 1500°C. Figure 5 illustrates the stability of some of the important ceramic packaging materials as related to temperature and the oxidizing-reducing atmosphere. As noted in Fig. 5, Al_2O_3 ceramics are very stable and may be heated to temperatures in excess of 1500°C in a H_2 ambient without destroying the alumina structure while keeping the refractory metals such as molybdenum (Mo) in their metallic state. This thermal stability has made possible the creation of a variety of processes for producing ceramic-metal composite structures of utility in packaging applications which will be discussed in more detail in Section III.

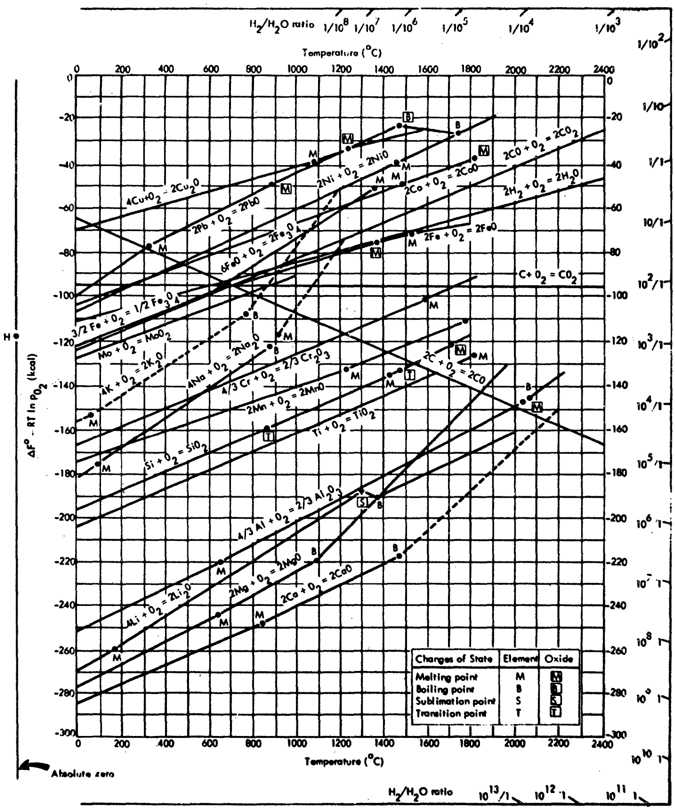


Fig. 5. Free energy of formation of selected oxides (after F. D. Richardson and J. H. E. Jeffes, J. Iron Steel Inst., 160, 261, 1948).

Mechanical

The strength and dimensional stability of ceramic materials make them particularly suitable for packaging applications. Because of the high modulus of elasticity and a virtual lack of yield, ceramics perform under the typical loads and temperatures required of the package without concern for dimensional distortion or shape changes. However, ceramic materials generally fail with a brittle fracture. This behavior must be carefully accounted for in design and handling to avoid yield losses due to chipping or outright catastrophic crack failures of the ceramic parts.

The brittle failure characteristic, plus the complexity of factors which can influence the ultimate strength of ceramic materials, has been a big factor in limiting the use of ceramic materials as structural members. However, this has not been a controlling factor in the application of ceramic materials to electronic packaging.

Typical mechanical properties of ceramic packaging materials are given in Table VII. It is noted that ceramic materials have a higher modulus of elasticity ($E \sim 10^7$ to 6×10^7) than other materials which is a result of the strong inter-atomic bonds. One desirable consequence of this high modulus referred to earlier is the low strain, therefore, good dimensional stability of ceramic packaging materials under load during typical service conditions. Organic materials have elastic modulii which are typically two orders of magnitude lower and consequently frequently require "stiffeners" to prevent deformation.

TABLE VII. Strength and Young's Modules for Ceramic Packaging Materials.

Material	Young's Modulus (E) ($x10^{-6}$ psi)	Flexural Strength (psi)
Al_2O_3	53	50,000
BeO	45	25,000
$MgAl_2O_4$	34	25,000
MgO	30	25,000
$ZrSiO_4$	30	22,000
$3Al_2O_3 \cdot SiO_2$	21	16,000
SiO_2 (glass)	10.5	15,500
Borosilicate glass	7.4	12,000

Based on inter-atomic forces, it can be shown that the fracture stress of ceramic materials should be approximately E/10 yielding theoretical tensile strength values in the range of 10^6 psi. Although correlation between strength and E is evident in Table VII, have been observed only in specially prepared glass fibers and in oxide whiskers in which defects have been avoided. The values reported in Table VII are typical and differ from theoretical due to the role of one or more of the following factors:

a. porosity and distribution
b. grain size, shape and distribution
c. secondary glassy or crystalline phases
d. surface cracks or flaws
e. internal stresses
f. dislocations

Chemical

The packaging material is subjected to a variety of chemically corrosive environments related to plating processes or soldering necessary to prepare the part for lead attachment or chip joining.

Most commercial polycrystalline ceramics used for packaging applications contain phases in addition to the main chemical species. Thus, alumina ceramics often contain as much as 10% secondary phases other than the grains of alumina and generally a significant portion of the secondary constituent is a glassy phase. The corrosion behavior of this glass often controls the chemical stability of the polycrystalline-ceramic, thus its corrosion characteristics are of importance.

Silica at low temperatures is practically non-reactive with water; therefore, most high silicate glasses show high resistance to moisture attack. However, alkali (i.e., Na_2O) containing glasses and non-silicate (i.e., B_2O_3) glasses have poor resistance to moisture attack. Further, the alkali leached from the glass increases the alkalinity of the water and this in turn increases the rate of attack due to the reactivity of silica to alkaline solution.

The alkali released by the glass during attack by acids tends to neutralize the acid diminishing the ratio of attack. Modifying oxides such as CaO, ZnO and Al_2O_3 are more soluble in acids. All glasses are attacked by hydrofluoric acid at rates controlled by temperatures and concentrations.

The secondary glassy phase constituent in most ceramics used commercially for packaging is generally low in alkali and may contain appreciable quantities of alkaline earths (CaO, BeO, MgO, etc.) and alumina and silica. Thus, moisture stability is not usually a problem and they are generally quite stable in solutions required for plating.

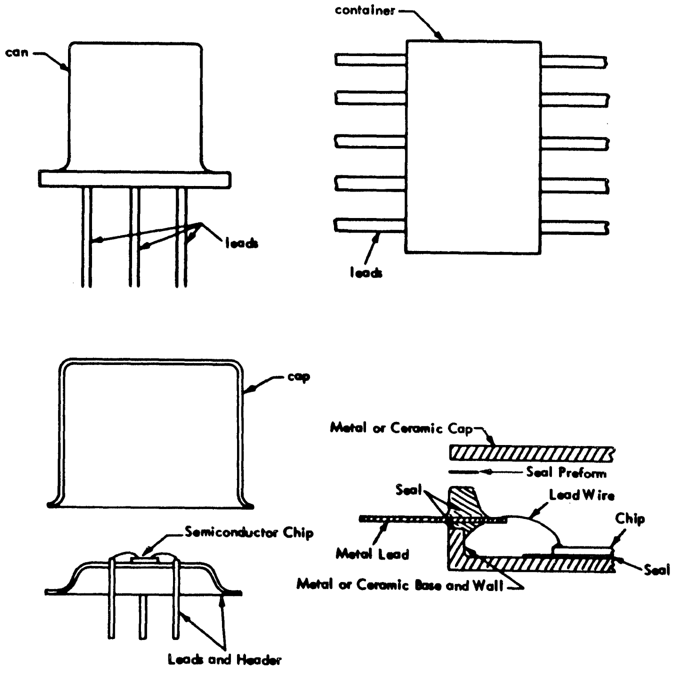
III. PROCESS

The use of ceramic materials in packaging applications can be categorized in two structures: 2-dimensional and 3-dimensional.

Two Dimensional

The two-dimensional ceramic substrate application encoumpasses the more familiar, prosaic, packaging concepts available commercially known as "flat pack", dual in line packages, TO headers and versions thereof. These structures seldom serve more than a fan-out function and as such discussion will be limited. These structures are shown schematically in Fig. 6.

The ceramic pieces required in the construction of these packages are prepared predominantly by compaction of the ceramic powders into the desired shape (dry pressing) followed by a high temperature process designed to cause the ceramic powder to coalesce and sinter dense creating a product with the desired ceramic properties. This is a well established process[2] and the ceramic vendor provides considerable assistance with regard to tolerances and dimensional capabilities.



a. TO-Type Package

b. Flat Package

Fig. 6. "Fan-out" packages.

Ceramic pieces which are generally less than .030" thick are generally prepared by a tape manufacturing process. This process is a precursor to the laminated ceramics multilayer process and will be discussed in greater detail later.

The two dimensional package has fundamental limits: maximum of two surfaces for wiring, limited functional capability (See Table I) and quickly I/O limited.

Thus, for application requiring high performance, these packages will give way to technologies which allow the use of the third dimensional (i.e. 3-dimensional structures) for increased functionality in the areas of wiring and voltages distribution and the like.

Three Dimensional

The "grid gap" referred to earlier has created the need to develop ceramic fabrication processes which utilize the third dimension of the package (substrates) for wiring purposes. The recent literature abounds with proposals and efforts to process materials to fulfill this need meeting a level of success which can generally be equated to the degree to which the total packaging problem was addressed. Those process approaches utilizing ceramics materials can be placed in two categories: batch process and sequential process.

Batch Process

The most common batch process for creating multilevel wiring structures is that which is often referred to as the laminated ceramic process. The fundamental steps in the process are listed in Table VIII. Many versions of the process have been described in the literature.[3-8] The casting process entails preparing a slurry or paint which contains the desired ceramic powder dispersed in an organic system which contains one or more of the following: a resin, wetting and dispersing agents, plasticizing agents and an organic solvent system. This paint-like slurry is cast onto a carrier and dried to yield a flexible film of the desired thickness.

The film (ceramic green sheet) is subsequently cut into sizes convenient for further processing as individual layers in the multilevel package structure. Holes are punched as required which correspond to interlayer intraconnection vias.

Next, the conductor metallization is deposited into these holes and by the use of screen printing technologies, metallization patterns are deposited on the surface of the green sheet layers representing a signal plane, ground or voltage plane or perhaps a translation (fan-out) function.

After inspection, the required layers are collated, placed in registration and laminated at low temperatures and moderate pressures utilizing the thermoplastic characteristics of the organic binder system of the ceramic green sheet to cause the indivisual layers to fuse together as one composite structure.

The composite structure is trimmed to the desired unfired dimension and then subjected to an appropriate high-temperature sintering process. During this operation the following occurs:

a. volatilization of organics used in the metallization pastes and the ceramic green sheet
b. densification of ceramics (elimination of pores)
c. metallization becomes conductive
d. ceramic-metal bonds develop

Following the sintering process, the resulting three dimensional wiring structure may be processed for lead (I/O) enclosure and chip attachment utilizing the methods developed for the earlier, less functional 2-dimensional ceramic packages.

Capabilities and Limitations

The laminated ceramic process can utilize virtually any ceramic material which is available in powder form. Ceramics for high performance packaging application, however, are selected such that they satisfactorily meet the needs shown in Table IX.

TABLE IX. Ceramic Selection Criteria.

Product Performance

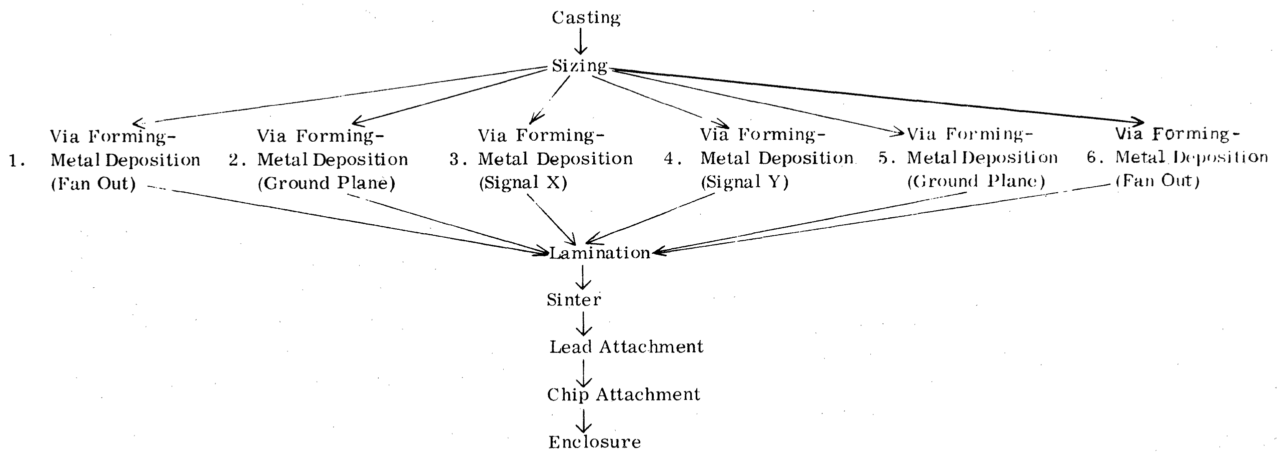
1. **low dielectric constant for signal transmissions**
2. **thermal conductivity for power dissipation**
3. **thermal expansion similar to silicon for direct chip mounting**
4. **mechanical strength to meet metal adhesion and handling requirements**

Process Performance

1. **thermal stability**
2. **chemical stability**
3. **reproducibility of properties**

Alumina (Al_2O_3) and beryllium (BeO) ceramics have fulfilled these requirements most satisfactorily to date and the reasons for this were dealt with in detail in Section II.

TABLE VIII. Multilevel Wiring Package - Batch Process Schematic.


Casting
Sizing
1. Via Forming-Metal Deposition (Fan Out)
2. Via Forming-Metal Deposition (Ground Plane)
3. Via Forming-Metal Deposition (Signal X)
4. Via Forming-Metal Deposition (Signal Y)
5. Via Forming-Metal Deposition (Ground Plane)
6. Via Forming-Metal Deposition (Fan Out)
Lamination
Sinter
Lead Attachment
Chip Attachment
Enclosure

The selection of the ceramic material places a significant constraint on the choice of the metallurgy. As noted earlier, the sintering process requires that the internal metallurgy and the ceramic be exposed to the same high temperature cycle. As most alumina and beryllium ceramics require temperatures in excess of 1400°C for densification, only the more refractory metals may be selected. Table X lists the characteristics of the more important candidates.

TABLE X. Characteristics Conductor-Metal Candidates.

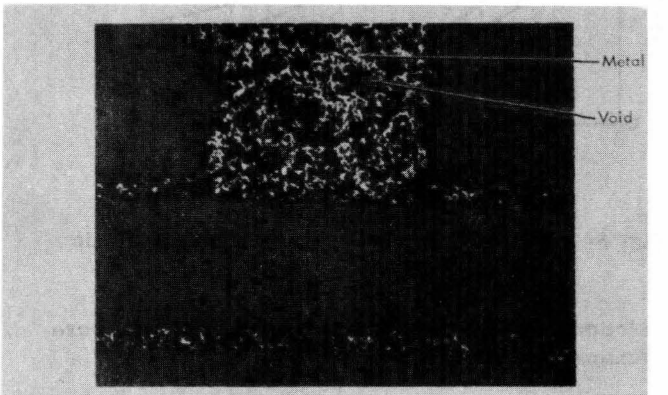
Metal	Melting Point °C	Boiling Point	Electrical Resistivity μohm-cm	Thermal Expansion $10^{-6}/^{o}C$	Cost ($/cm³)	Sintering Atmosphere Consideration
Silver	961	2210	1.6	19.1	.64	air
Gold	1063	2970	2.2	14.2	27.70	air
Copper	1083	2600	1.7	17.0	.01	reducing
Palladium	1552	4000	10.8	11.0	13.20	air
Platinum	1769	4410	10.6	9.0	123.00	air
Rhodium	1960	4500	4.7	8.5	55.40	air
Iridium	2442	5300	5.3	6.5	335.80	air
Molybdenum	2610	4800	5.2	5.4	.10	reducing
Tungsten	3410	6700	5.5	4.5	.23	reducing

The data in Table X highlight several constraints with regard to the selection of metallurgies for use with alumina and beryllia ceramics:

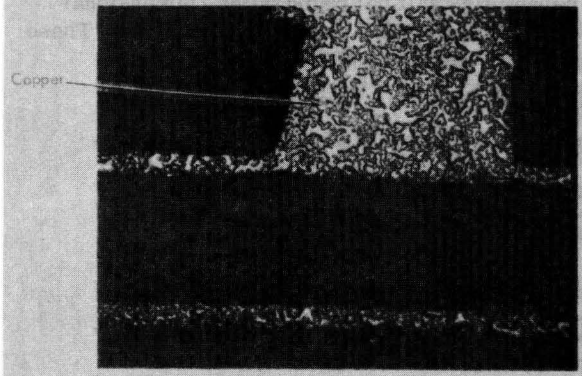
1. Elevated sintering temperature (> 1400°C) preclude the use of high conductivity Ag, Au or Cu in the cofiring process. (This handicap has been circumvented by the capillary-fill metallizing process described later).
2. W and Mo, the next most desirable candidates (W, Mo,), electrically, require reducing atmospheres (H_2) during the sintering operation to prevent the metal from oxidization.
3. Pt, Pd, lr and Rh may be used allowing the use of air sintering equipment if the sacrifice in conductivity and increased metal costs are not critical.

Thus, because of the characteristic of the metals available, two options are available for the sintering of laminated ceramic structures: the use of the noble metals (Pd and Pt), which make it possible to sinter in air; or the use of the refractory metals (Mo, W, etc.) which dictate the use of reducing atmospheres to prevent oxidation of the metals. The criteria for selecting the noble metal versus the refractory metal approaches is not clear cut to date and is evidenced by the fact that major vendors are pursuing both routes. In general, the major advantage of noble metal systems is their compatibility with the vast thick film technology (resistor, capacitor, etc.). The refractory metal systems provide lower conductor resistivities and are compatible with the ceramic-refractory metal seal technologies which have led to strong ceramic-metal bonds with proven hermeticity.

An internal metallizing process has been developed to overcome the conductivity limitations imposed by the ceramic-metal co-firing process referred to earlier. This process,[3] has made it possible to achieve conductors within Al_2O_3 ceramics with conductivities approaching 60% that of bulk copper. The method is similar to the laminated ceramic process with the exception that the co-fired W or Mo metallization is selected so that following the sintering process, a lightly metallized porous channel is formed. This porous channel is subsequently filled with molten copper using capillary forces. Structures before and after filling are shown in Fig. 7.



a) Before Capillary Fill



b) After Copper Capillary Fill

Fig. 7. Copper-filled conductor (240X).

Figure 8 summarizes the resistivities which have been observed to date for the various metallurgies compatible with the laminated ceramic metallurgy process. Data for pure copper is given for reference purposes.

The sheet casting, hole (via) forming, and screening processes have created a capability, coupled with the properties of the ceramics, which provides a meaningful approach to bridging the grid gap referred to in

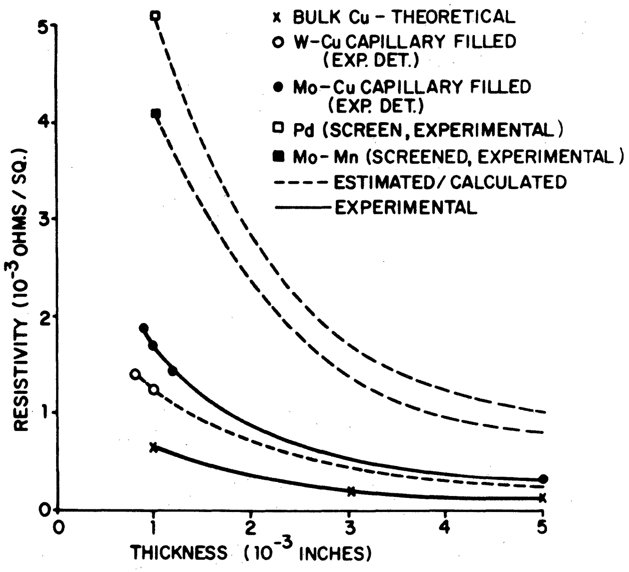


Fig. 8. Electrical conductivity of laminated ceramic metallurgies.

Section I. The general capabilities in this regard are summarized in Table XI.

It should be emphasized that the capabilities listed do not represent known fundamental limitations. They represent state-of-the-art capabilities. The fundamental process limitations which will ultimately limit design capabilities will be: registration, dimensional control and metal-ceramic diffusion/interaction. These limits have not been determined as yet.

TABLE XI. Batch Fabricated-Laminated Ceramic Design Considerations.

Parameter	Standard	Possible
Conductor Width	8-10 mils	4-5 mils
Clearances	8-1- mils	4-5 mils
Via Hole Diameter	10 mils	4-5 mils
Via-Center to Center	25-30 mils	10 mils
Interconnection via Density	1.6×10^3 vias/in^2	1×10^4 vias/in^2
Dielectric Layer Thickness	5-7 mils	2-up (multiples)
Number of Functional Layers	5-10	unknown
Overall Size	.5"x.5 to 3"x3"	~ 6" x 6"

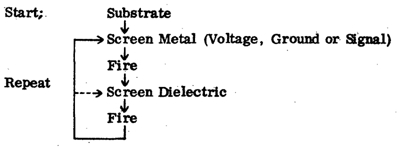
Sequential Process

Three-dimensional wiring arrays utilizing ceramic materials are being fabricated by a variety of methods categorized as sequential processes. That is, individual functional layers are built up on a substrate in a sequential process mode.

The sequential processes have generally utilized thick film processes,[9,10,11,12] however techniques employing photolithographies printing[13] and evaporation-vapor deposition[14] processes for simple structures have been reported. These latter processes are recent developments and will not be discussed here.

The thick film sequential process schematic is shown in Table XII.

TABLE XII. Thick Film Sequential Process for 3-Dimensional Wiring Structures.



The thick film deposition process entails extruding a paste containing the desired metal or dielectric through a metal mask onto the substrate. The resulting deposit is dried and then subjected to an elevated temperature designed to sinter the metals and dielectric-insulator material dense. The cycle is repeated to yield a multilevel structure with the desired number of layers.

Capabilities and Limitations

The metal and dielectric materials selected are generally proprietary compositions. The metals are selected from those listed in Table X, while the dielectrics are selected from the following classes of materials: ceramic (high-and low-dielectric constant) glasses and glass-ceramics.

As in the laminated ceramic process, materials are selected based on their process compatibility utilizing the materials characteristics enumerated in Section II. Glasses and glass-ceramics are commonly selected as they can be densified at temperatures below 1000°C utilizing fairly simple furnace equipment, however, alumina refractory metal structures are also common.[15]

The lower melting metals, i.e., Ag and Au, (see Table X) are used with the low temperature dielectrics.

General capabilities of the thick film sequential process are given in Table XIII. The design capabilities listed are quite similar to those given for the laminated ceramic process in that both technologies rely on screen printing processes. However, there are some significant differences worthy of note:

1. Via hole diameters of less than 10 mils are difficult as they tend to close during the firing process.
2. Individual dielectric layers thicker than 1-2 mils are difficult due to the limitations of the screen printing process.

TABLE XIII. Sequential Process Thick Films Design Consideration.

Parameter	Standard	Possible
Conductor Width	8-10 mils	4-5 mils
Clearance	8-10 mils	4-5 mils
Via Hole Diameter	15-20 mils	10 mils
Via-Center to Center	30-40 mils	20 mils
Interconnection via Density	1×10^3 vias/in^2	2×10^3 vias/in^2
Dielectric Layer Thickness	1-2 mils	1-2 mil (multiples)
Number of Functional Layers	3-4	4-7
Overall Size	.5"x.5" to 3"x3"	~ 6" x 6"

3. The number of functional layers is limited by the accumulated topography effects and the camber due to expansion mismatches of materials deposited.
4. The sequential aspect of the process dictates a need for high yields at each process step in order to minimize costs.

The sequential thick film three-dimensional wiring structure will augment the batch fabricated ceramic structures in the following areas:

1. Making 3-dimensional wiring structures requiring few layers and low part numbers where the capital expense of the laminated process might be prohibited.
2. Modeling of new simplified design structures prior to committing the structure to the tooling expense required of the laminated ceramic process.

SUMMARY

The integrated circuit packaging challenge can be described as a grid gap which exists between current semiconductor technologies and present printed circuit board capabilities. Meeting this challenge requires the development and use of materials which can fulfill the multiplexity of inter-material reactions and constraints required of the composite metal-dielectric structures. To accomplish this the materials must satisfy electrical, thermal, mechanical, and chemical properties dictated by processing needs and the product performance requirements.

Ceramic materials present a broad range of properties which have contributed to their success in addressing the packaging needs. Further new processes have been developed which have made it possible to bridge the grid gap which exists between the semiconductor chip terminals and the organic printed circuit end.

REFERENCES

1. Kittel, C., Intrdocution to Solid State Physics (Wilby & Sons, Inc., New York, 1966), 3rd Ed., p. 385.
2. Kingery, W. D., Ceramic Fabrication Processes (MIT Technology Process and Wiley & Sons, Inc., New York, 1957), Chapter 7.
3. Schwartz, B. and Wilcox, D. L. "Laminated Ceramics", Proceedings, Electronics Components Conference, 1967, Washington, D.C., p. 17.
4. Fehr, G. K., "Microcircuit Packaging and Hermetic Sealing", Hybrid Circuits Symposium, WESCON, 1968, Los Angeles, Calif.
5. Fagersten, E. G., "Multilayer Ceramics Interconnect Substrates-Dielectric Substrate Integration", Hybrid Microelectronics Symposium, 1968, p. 401.
6. Keller, W., Pirigyi, F., Cole, G. and Budd, J. P. "Multilayer Alumina Circuit Boards", Proceedings Electronic Components Conference, 1969, Washington, D.C. p. 52.
7. Theobold, P., Davis, M., Bailey, J., "Multilayer Composite Substrates for Thick Film Hybrid Microcircuits", Hybrid Microelectronics Symposium, 1969, Dallas, Texas, p. 447.
8. Shahbender, R., Wentworth, C., Li, K., Hotchkiss, S., and Rajchman, J. A., "Laminated Ferrite Memory", Proceedings - Fall Joint Computer Conference, 1963, p. 77.
9. Fenster, H. and Davis, M., "Multilayer Ceramics for Interconnecting IC's", Electronic Products, May 1968, p. 60.
10. McCormick, J. F., and Calabrese, D. W., "Multilayer Microelectronics Substrates, A Materials Evaluation", Proceedings of the Technical Program, NEPCON 1969, Annaheim, Calif, p. 500.
11. Stein, S. J., "Glass-Ceramic Glazes for Crossover and Multilayer Screened Circuitry", Proceedings, Electronics Components Conference, 1968, Washington, D.C. p. 119.
12. Moore, J. R., "A Four Level Screen Printed Multilayer", Micro-Multilayer Interconnection Substrates Seminar, NEPCON-East, 1969, Phila., Pa.
13. Duffek, E., "An Electroplating Process for the Fabrication of Double Layer Metallized Ceramics", Tech. Session II, NEPCON-West, 1970, Annaheim, Calif.
14. Technical News Item, "Interconnecting Via Multilayers", Electro-Technology, Nov. 1969, p. 16.
15. Frederick, J. D., "Economic Considerations for Micro-Multilayer Interconnection Substrates", Micro-Multilayer Interconnection Substrate Seminar, NEPCON-East, 1969, Phila., Pa.

TRADEOFFS IN PACKAGING BEAM LEAD INTEGRATED CIRCUIT SYSTEMS

M. T. Skubiak and D. J. Wadsworth
Bell Telephone Laboratories, Inc., Naperville, Illinois

ABSTRACT

Partitioning considerations based on processing yields, circuit interaction, fault recognition and diagnosis are explored with emphasis on the interdisciplinary problems of integrated circuit system physical design. These are weighed against overall system cost and reliability of a large, real-time digital system.

THERMAL DESIGN OPTIMIZATIONS IN LARGE DIGITAL COMPUTERS

R. C. Chu, J. H. Seely, V. W. Antonetti and A. L. Pascuzzo
Systems Development Division, International Business Machines Corporation
Poughkeepsie, New York

ABSTRACT

This paper addresses the thermal problems encountered in the development and design of large digital computers. It first reviews the state-of-the-art of thermal design techniques. Thermal designs for several well-known large digital computer systems will be used as illustrative examples.

The second part of this paper discusses the thermal system design criteria. It outlines all pertinent factors affecting the overall rating of a thermal system. Emphasis will be placed upon trade-off considerations and optimization techniques. Among other things, it will present a comprehensive treatment of coolant selection and thermal environmental considerations.

The third part of this paper will concentrate on the discussion of special thermal techniques for local thermal control or hot spot cooling. Thermo-electric and heat pipe techniques will be demonstrative examples among others.

The last part of this paper will present a summary of performance comparison of today's thermal systems; air systems, air-liquid systems, and liquid systems. This paper will be concluded with a projection of future requirements in thermal design of large digital computers.

PACKAGING ASPECTS OF THE UNIVAC 9200 DESIGN

W. F. Simon
Univac Division of Sperry Rand Corp.

ABSTRACT

The design of a small computer system for business data processing is as sensitive to the choice of packaging techniques as to the selection of memory cycle time or logic gate speed. The UNIVAC(R) 9200 was designed for volume production with complete freedom for innovation in all basic packaging parameters. The choices made and the technical justification of such choices will be explained.

There are two kinds of goals in the world of Digital Computers. One is to design a system so powerful that it can accomplish tasks which have thus far been impossible - or at least impractical - to complete. The other is to design a system so simple to build, use, and maintain that it becomes possible to extend the application of data processing to areas in which costs have previously been prohibitive.

This is not to imply that there isn't a large market between these extremes. In fact, very substantial income is derived from the existing market of experienced users, but the ability of a system to open new markets is the hallmark of success.

UNIVAC(R) has done well on both sides of this street, being fortunate to have the substantial resources needed to supply the full range of equipment needs and substantial software support as well. The challenge of building greater performance than any existing design can offer is the easiest to understand, and is limited only by the daring of the designer. The goal of the simple machine design is far less dramatic and yet requires a level of skill, insight, and innovation quite uncommon in the engineering profession.

The UNIVAC 9200 design represents a successful attack on the economic frontier of the computer world of the mid-1960's. It may seem clumsy in the LSI world of the 1970's, but it provides an excellent example of the importance of many small but related decisions.

In planning this system the basic dilemma was to provide a minimal system which could stand alone and yet retain the potential for growth to a performance level an order of magnitude greater. On one hand, nothing can be wasted in the minimal system; but on the other hand, each step upward to greater performance should be easy to make. Some considerable experience with this problem had taught us that both the minimal and the expanded design must be considered as a unified problem, and that success in designing a profitable minimal system would go a long way toward providing a highly competitive design throughout the desired range of performance.

What are the absolute minimum requirements for a viable business data processing system?

It must process data files. These can be perforated paper tape, punched cards, or magnetic tape. Since business applications are commonly associated with punched cards, the minimal system was oriented around a card reader and a card punch.

It must store and retrieve data and instructions rapidly in any arbitrary sequence. For acceptable performance, some minimum amount of random access storage capacity must be established.

It must prepare printed reports. The minimum system must therefore include a printer of some kind.

There must be some kind of control panel to initiate, monitor, and terminate operations.

And, of course, the minimal system must include the arithmetic and control circuits of the central processor.

Although the design and packaging of the mechanisms used for input and output of digital data represent a substantial portion of the 9200 System Design, we will pass over these problems with only a few general comments.

UNIVAC(R) is a registered trademark of the Sperry Rand Corp.

1. In most cases, mechanisms are housed in cabinets which are quite wasteful of useful volume.

2. Signals required to control mechanisms are commonly large in voltage and current levels; the responses returned from them are usually small, weak, and noisy.

3. Mechanical maintenance accounts for the lion's share of the field engineer's work on a small computer system. This favors the man who is a mechanical genius but a little fuzzy on electronic and logical concepts. When a failure occurs in the electronic segment of the system, it will take a long time to isolate and repair unless considerable planning has gone into the preparation of maintenance aids and fault isolation procedures.

These considerations had considerable impact on the basic decisions concerning 9200 electronic components. In the first place, we wanted a small system that looked like a small system. The considerations of operator convenience and mechanism accessibility made it clear that a minimum system would occupy a volume which could easily swallow a substantial segment of the electronics. We could avoid the over-shadowing bulk of a five foot high "electronics cabinet" and simplify intercabinet cabling if we made efficient use of the volume already available under the 40 inch high "work surfaces" of the various mechanisms.

This meant, however, that basic logic elements would have to co-exist with high powered solenoids. It also meant that volumetric efficiency of electronic packaging would be important, and that the problem of cooling the logic gates would suffer from the restricted space.

An even more serious consideration was the realization that efficient use of space would eliminate the use of duplicated printed circuit assemblies in the logic module. In "second generation" computers such as IBM 1401 or UNIVAC 1004 a few basic cards were used repetitively to supply most logic functions. When one card was suspected of malfunction, another could be found in an operating logic section for a comparison test. If interchanging the cards moved the malfunction, the fault was clearly isolated. This kind of trouble shooting requires a minimum of skill, which is an advantage more significant than other more obvious factors, such as reduction of spare parts inventory and increase of manufacturing volume on favored assemblies.

The difficulty is that it requires nearly all logic connections to be made via the card connector, so that a single component assembly may perform a wide variety of functions. Thus, any increase in volumetric efficiency of logic packaging would be limited to the practical improvement possible in connector pin density. At most, the number of connector pins per unit area of backpanel had only increased by a factor of four while electronics shrunk from discrete component technology to integrated circuit technology.

UNIVAC$^{(R)}$ decided in favor of higher density packaging, with increased support in the development of maintenance techniques to compensate for the loss of package duplication. The 9200 logic averages only one duplicated card in every seven card types used for basic logic (excluding such things as solenoid drivers, where power density forces lower component density). But the system is also supplied with a fault dictionary and a detailed sequence of fault isolation instructions. By following this carefully chosen sequence of simple operations, and recording the resulting digital codes (usually displayed on the operator's panel), a fault symptom is generated. These symptoms are sequentially listed in the dictionary, and the malfunction is identified with a printed circuit card location.

Obviously such a technique is aided by functionally grouped logic elements on each card. It is easier to identify an adder error and replace the entire block of adder logic than to identify the specific gate at fault. This is consistent with the goal of increasing logic density, which also profits from functional grouping of components.

The card connector chosen for UNIVAC 9200 printed circuit assemblies is a design originated for use in 1108. It has 55 terminals in an area of 1½ square inches. To keep the insertion and withdrawal forces low without loss of contact pressure we literally put "a square pin in a round hole". The "hole" is a split sleeve formed in a progressive die, which gives two independent spring elements shaped to make four lines of contact with the corners of the square pin. The sharp corners of the square pins create a large normal force on a small area, assuring a gas-tight seal without excessive friction forces.

The square pins simply continue through the connector block and provide appropriate terminals for wire wrap on the opposite side. A simpler male connector pin is difficult to imagine. The female half of the connector "floats" in a molded cavity for easy connector mating.

It is provided with a tang which bends down and solders in the printed circuit board at a distance great enough not to restrict the contact "float". This part is somewhat more expensive, but provides extremely high reliability.

The choice of a basic logic gate design for the 9200 was not an easy decision. Up to 1965, monolythic integrated circuits had not been cost competitive with hybrid chip technology or even with the use of integrated resistor networks to couple discrete semiconductor elements. An extensive survey of the semiconductor vendors surprised us by showing a significant cost advantage in most monolythic gate structures.

The only structure for which a monolythic version was distinctly poor in both cost and performance was a simple diode gate, without gain. The 14 lead integrated circuit packages had an obvious analogy to the 16 or 18 terminal printed circuit cards of the computer families popular in the early 1960's. We knew, therefore, that a simpler set of basic logic functions could be designed with two levels of gating - the first level having only diode gates, and the second level providing a complementary gate function plus power gain and signal inversion. Using this philosophy, we defined a set of integrated circuit NOR gates which could be coupled with simple diode AND gates.

In order to be sure that this unorthodox approach was justified, an existing processor design was studied to establish a basis for comparison of the different logic families. The original design had been expressed in two level logic, suitable for the diode-coupled NOR circuits which we proposed. To obtain a fair comparison we re-designed a segment of approximately 3000 to 4000 gates to take advantage of Emitter Coupled Logic which was available "off-the-shelf". When this proved unattractive, another attempt was made, using Complementary Transistor Logic features - another "off-the-shelf" item. Equivalent counts of components in each of the logic families were determined and comparisons of cost and performance were made.

Both ECL and CTL suffered from incomplete use of the package terminals. Although each circuit was available in a 14 lead package, as few as six leads were actually used for logic interconnections on some configurations. This was somewhat a function of excess dissipation, but also reflected the use of extra voltage and ground pins. The net result was less function per package, and hence more packages required per system. This more than offset the lower cost of a standard package as compared to a custom design.

The use of diode coupling between NOR circuits would not have so easy a time today, since the value of an integrated circuit terminal is generally appreciated. Nonetheless, the 9200 logic elements retain significant advantages. On the average, each integrated circuit has 17 diodes associated with its input terminals. If the typical chip has three outputs, we have the equivalent of an integrated circuit with 20 terminals instead of 12. In addition, a great deal of flexibility is obtained since the diodes are connected only where needed. Many different functions can be accomplished with relatively few kinds of parts.

Note also that the diode gates dissipate very little power. Thus the complex 20 terminal logic function has virtually the same power dissipation as the simpler 12 terminal configurations commonly used as an integrated circuit building block. Diode coupling has not only reduced the average cost of the logic functions; it has also made it easier to cool.

Provision of adequate cooling also has a bearing on the general layout of the 9200 System. A small system should not require too many forced air paths. Blowers are part of that "irreducible minimum" of cost which every system must contain before the first useful function is installed. As the system grows smaller, this base becomes more important. To keep the number of air paths at a minimum, all printed circuit assemblies were placed in one of two card modules - the memory module, or the central processor module. Thus, only two blowers are used in the minimum system. (I am excluding the small axial flow fans which are an integral part of the power supplies, and a small centrifugal fan which cools the printer actuators, because these requirements are independent of circuit packaging).

Why not go all the way, and merge the memory module with the processor module? This was considered, but was rejected as too dangerous. The 9200 was the first commercial use of a plated wire memory, and fully enough problems were anticipated without adding cross-talk from the high powered circuits associated with the printer and the punched card equipment. In fact, we felt quite brave in merging logic circuits with the I/O control circuits. To make this practical, we specified our custom NOR circuits with an extra measure of noise margin.

Where the usual DTL or TTL circuit must overcome two silicon diode thresholds to turn on, the 9200 logic element must overcome four diode thresholds. One of the extra diode thresholds is wiped out by the forward drop of the coupling diode, but the remaining noise margin permits a noise spike approximately 3/4 of a volt greater than the worst case noise limit of the more conventional circuit configurations.

This would not be practical if we did not move the power supply side of the logic waveform as well. Use of a six volt power supply instead of the usual five volt source increases dissipation somewhat, but not to a level which will offset the low power dissipation of the diode gate structures. In fact, the internal circuitry of the chip could be designed with current sources whose variation between OFF and ON conditions is less, since a larger source voltage is available.

Experience and measured tests have shown that the 9200 logic is even less susceptible to interference from external noise sources than we had expected. A standard signal generator with a 41 inch whip antenna three feet from the cabinet could not deliver enough power to disturb it. Severe line voltage transients and conducted voltage pulses applied to the cabinet structure caused no errors. This is scarcely surprising when one discovers that printer actuator drive circuits supply 170 volt pulses in the same wire-wrapped backpanel used by logic signal connections. As might be expected, some considerable care of wire routing in the vicinity of these high powered circuits, and careful attention to ground return paths were necessary to accomodate this circumstance.

Having chosen to package all electronics except the memory in a single area, there still remained a choice of selecting which area. The printer thrust itself upon us. It requires a heavy bundle of wires to couple actuators with their driving circuits. Up to a reasonable distance, these cables can be twisted pairs; beyond this coaxial construction would be needed. The cost increment would be dramatic. Therefore, the central processor found its home inside the printer cabinet.

Merged with the central processor card library we now find the printer control logic and associated drivers, sense circuits for paper motion and character sequence, and also its cable connections. Rather than wire a rack-and-panel type connector to the card library, we simply solder all cable connections on a printed circuit board and plug it directly into the wire-wrapped back panel.

Similarly, the card reader logic circuits and the card punch controls, with their associated photocell signal amplifiers and power drive circuits are part of the main card library. Cables branch out from the card insertion side of this module to connect the mechanisms. Although separate cabinets house the reader and punch mechanically, very little of the associated electronics can be found in these cabinets. Hence, no cooling is required, and no separate frame to mount replaceable cards is needed.

Substantial cable lengths are used, but these are made tolerable by providing that all cabinets bolt together in an L-shaped floor plan. Cables simply pass between such cabinets through holes in the adjacent sides. A continuous cabinet frame, with all sections electrically connected and grounded, provides a crude electrical shield. Where signals are small, separate cable shields and ground return paths (tied to frame ground at only one point in the system) maintain a low noise level.

At the corner of the L, there is a cabinet whose accessibility is rather poor. From the operator's station, only a six inch segment is exposed. At the rear and on the right side there are removable panels, but if one is chosen as a means of access, the other becomes somewhat redundant. A good part of it faces the side of a module whose front is the maintainable part.

In this cabinet we chose to mount the plated wire memory. It is accessed from the rear of the installation and this is the prime access to the cabinet. On the side, however, one has limited access to the volume behind the memory module. At this point we mounted the main system power supply. It requires little attention, and it is acceptable to slide it completely out of the cabinet for service. Above this supply the controls for power sequencing and protection are mounted. Access is provided by lifting the table top, except that the circuit breakers may be operated from the side.

Grouping the memory and the power supply in a common frame was a fortunate choice. Power requirements for the memory are more complex and more critical than for any other need. When memory expansion requires a duplicate module to be added, it is simple to roll up a second cabinet almost identical to the first, with the same simple power supply and a "slave" power sequence unit. Some power supply capacity is wasted, but not enough to justify a second design.

The UNIVAC 9200 processor was designed to take full advantage of the high speed and low cost of the plated wire memory. This was done primarily by letting a small part of the memory serve as the working registers. In the usual machine organization, these registers are flip-flop aggregates and comprise a significant portion of the processor cost. Not all working registers can be included in the memory; for example, the memory address register must be external in order to access memory. However, the 9200/9300 can carry this concept further than other systems while retaining a significant speed advantage because of the memory speed. Only 29 bits of flip-flop registers were required for the arithmetic and control portions of the processor. A total of 144 bits of control register storage are contained in the memory. In addition, a dual set of arithmetic and indexing registers are contained in the memory, providing a zero interrupt acceptance time. This feature is important in achieving maximal throughput rates since interrupt acceptance time effectively decreases I/O device speeds. This feature is believed to be unique for machines of this low price class.

To meet the cost goals of the 9200 system design a substantial reduction in manufacturing cost of the control panel was necessary. This was achieved by a unique combination of permanent magnets and reed switches which replaced the usual mechanical control panel switches.

Commercial computer control panels have traditionally been built with mechanical switches having either directly exposed contacts or, at best, covered for dust protection. Airborne contaminants, such as solder flux vapor, ultra fine particulate matter, and the normal atmospheric content of sulphur and oxygen, have all served to lower contact life and necessitate costly replacement or mainenance procedures.

Where neatness was important, the wiring connecting the switches had to be laboriously measured and formed into tied cables. Subsequent stripping and tinning of the wire ends preceded the often delicate job of hand soldering the wires to the many switch contact lugs. Errors in making the cables and marking its wires, mistakes made in wiring to the wrong contact lugs, poor soldering which led to cold joints, shorted switch contacts, and burnt insulation were common difficulties. These recognized manufacturing faults were controlled by setting up, at significant cost, quality control monitoring, testing, and rework procedures.

At the start of the 9200/9300 development program, it was proposed that glass reed capsules could produce a control panel that would eliminate many of the faults of the designs then in use. The immediate and most obvious advantages were:

1. The glass reed capsules themselves would be immune to atmospheric-borne contact contaminants. This in itself would completely eliminate any necessity for field procedures pertaining to contact cleaning.

2. Mounting the capsules as components on printed wiring boards would allow conventional dip soldering to be used to eliminate the delicate hand soldering as well as the cable making, stripping, tinning, and securing operations.

Briefly, the switch is formed as follows: A rocker-arm containing a permanent magnet is inserted into a plastic holding cavity which in turn is inserted into a control panel opening where it snaps in place. The rocker-arm is held and allowed to pivot in the cavity by a coil spring and either one of two spring retainers, the choice of retainer being based upon whether "momentary" or "alternate action" operation is desired. Depressing one end of the rocker-arm moves the magnet over the overlapped ends of the reeds, causing them to make contact. The reed contact is broken by depressing the opposite end of the rocker-arm, which moves the magnet away from the reeds.

Several major design efforts were necessary before a reliable magnetic switch could be produced. One of these efforts was to determine the aging characteristics of permanent magnets to ensure long term reliability. The resulting study was so thorough and unique that the magnet vendor requested permission to include the test results in a textbook being written on permanent magnet theory. Other design efforts were necessary to determine the proper mounting angle for the magnets in the rocker-arm that maximizes the manufacturing tolerances, and to incorporate a mechanical and magnetic "snap" action feel essential for operated switches.

This work has provided all the hoped for advantages. Manufacturing cost of a 9200 Control Panel is about 1/5 the cost of similar panels in earlier UNIVAC systems. Maintenance is simple and infrequent.

The plastic rocker arm seems more likely to fail than the reed capsule. In either case, repair is the replacement of a component; no adjustments are needed.

It may be that these details are rather far afield from conventional concepts of packaging design. Perhaps it does make a point, however, that bigger and faster machines do not have all the fun. Bringing the computer to the small business man can be challenging, and the audience is both large and enthusiastic.

HOW MUCH LOGIC SHOULD BE PUT IN A DISPLAY TERMINAL?

Ivan E. Sutherland

Evans & Sutherland Computer Corporation

The rapidly changing economy of digital logic forces display designers to reconsider the trade-offs between logic and memory. Now that we can get substantial logical capability in our display systems, the crucial design question is:
"What shall we use the logic for?"

There are two kinds of logic that can be built into display terminals. On the one hand, there is logic which interprets compact notations in memory and expands or processes them into pictures on a Cathode-ray tube. On the other hand, there is logic which interprets operator action to modify the picture information as stored in memory. A good terminal design, in my opinion, should include a good deal of wired logic for interpreting picture data and producing pictures. Such a terminal design can capitalize on the relatively low cost of logic to economize on the amount of memory required for storing the picture or to improve the logical structure of the information storage. Such systems can make it very easy for a programmer to set up and modify complex picture divisions.

On the other hand, it is not good design to include in a terminal wired logic which interprets operator action. Flexibility in the use of display systems requires that a programmer be free to re-define the meaning of any operator action. The interpretation of what the operator means when he presses a certain button should be left to stored program logic.

It has been my design philosophy for some years to couple display terminals as closely as possible to the memory of their host computer. Such display terminals behave like stored program computers whose op-code set is specialized to display functions. These special purpose display processors communicate with the general purpose computer to which they are connected. The communication between host computer and display is, thereby, simplified.

An alternative design widely used elsewhere is to separate the display refresh memory from the host computer. At best, such a separation forces the host computer to prepare information in blocks for transmittal to the separate memory. At worst, such a separation imposes a very narrow bandwidth communication channel between the host computer and the display system. Separation of the display refresh memory cycle demands of refresh or, if the display system is to be placed so far distant from the host computer that narrow bandwidth communication is required. In either case, the responsiveness of the system will be less than could be achieved by a design using communication in a common memory. Most modern computers have multiple port memory systems with many more memory cycles available than are required by the central processor.

THE USE OF VIDEO TECHNOLOGY TO PROVIDE ECONOMICALLY ATTRACTIVE GENERAL-PURPOSE GRAPHICS

T. O. Ellis, The RAND Corporation

Cathode Ray Tube (CRT) graphic displays offer one of the most promising communication paths between man and the computer. The significant advantages of displaying alpha-numeric information on the graphic consoles include the speed at which several hundred characters can be displayed, the great flexibility and lack of noise in presenting data, and the fast and quiet erasure. For the most esoteric problem, where dynamic recognition of events is not otherwise descernible, the graphic display offers the greatest hope for new levels of problem-exploration and problem-solving capability. By means of graphic display techniques, the user often can view a scene, structure, or relationship on a display in such a way as to gain new insight about his problem.

The challenge of a graphic display, suitably augmented with flexible input device capability, is how to supply terminals at sufficiently low cost to a community of concurrent on-line users; in a form where the user's terminal is "personal;" and where the terminal is the general communication instrument between the user and most of his computer needs. The RAND Video Graphic System is an experiment directed at one implementation of these aims.

RAND's system applies recently developed advances in analogue scan conversion and recording techniques to allow high centralization and fast multiplexing of the expensive portion of full graphics displays. It provides minimization of "console site" equipment to allow "ownership" of terminals by individual users.

The Video Graphic System is one approach to general-purpose graphics. Clearly, the system is only applicable where several users exist in one physical environment. The console cable length between the central hardware and any station is 2000-3000 feet. The VGS can be configured to serve several clusters of users, with each cluster being physically separated by several miles. In this case, each cluster would have a local disc buffer and a single microwave channel or coax between the central hardware and the cluster point.

In a world that is becoming digitally oriented, the analog nature of the VGS hardware seems a bit out of place. However, to supply full grey scale, full graphic capability, and other VGS features, the shared use of centralized analog techniques is the most economical approach that current technology can supply. It appears as if analog video technology will provide a economic advantage for at least the next few years.

CONSOLES: INTELLIGENCE AT THE MAN SITE

Glen Culler, Culler-Harrison Company

The quality of communication between the user and his system is of first importance in on-line system design. An illustration of the consequences to be achieved through high performance user consoles is presented.

The use of consoles of widely varying power and computational efficiency has become a most significant factor in the design and utilization of on-line systems. Our presentation will consist of live utilization of an on-line system showing the use of:

1. A display oriented Engineering Analysis System.

2. The telephone tied to a speech analysis and synthesis system using voice I/O to control the Engineering Analysis System.

Because of the importance of the operational dynamics and audience stimulated discussion, it is not possible to prepare a complete paper in advance.

THE COMPLEXITY OF GENERAL-PURPOSE REMOTE COMPUTER TERMINALS

Joseph F. Ossanna, Bell Telephone Laboratories, Inc.

ABSTRACT

Today's family of remote computer terminals exhibits a wide range of complexity and cost. Many currently advocated terminals fall at the high end of the complexity spectrum. The relationship between terminal complexity and utility is examined for general-purpose terminals. It is observed that some of the more complex terminals provide marginally greater and sometimes less utility, often at greater cost, than some less complex terminals. A desirable evolution in terminal design is suggested.

INTRODUCTION

The remarks that follow are meant to apply to the general-purpose class of remote computer terminals; that is, terminals that could be used interactively with various time-sharing systems by a wide range of users for a variety of applications. The main concern is the relationship of complexity to utility. There are several dimensions of terminal complexity. The complexity of design and manufacture usually increases with increased capability and influences cost. The complexity of use may or may not increase with increased capability and is a human engineering issue. Finally, the complexity of the hardware and software interface between the computer and the terminal influences the extent of support of the terminal and affects the cost of accessing the computer.

COMPLEXITY AND UTILITY

It is certainly true that the cost of a terminal generally increases with increasing complexity of manufacture (although such technology as large scale integration can reduce the rate of this increase). What is the relationship between the complexity of design and both the complexity of use and the general utility of a terminal?

Perhaps the simplest computer terminal in use today is the TOUCH-TONE (Registered Trade Mark of the Bell System) telephone; it provides a very small input character set and only audible output. Although TOUCH-TONE telephones are successfully being used in certain applications, their very simplicity makes their use awkward at best for many kinds of interactive computer usage. The inputting of any reasonable character set requires complicated multi-character mapping. The audio response must be short enough for ease of comprehension. It is difficult to imagine editing a FORTRAN program with such a terminal. It seems clear that a more complex terminal would be easier to use for general interactive computing.

The first notable step toward increased hardware complexity is typified by the Model 33 Teletypewriter which costs perhaps an order of magnitude more than a TOUCH-TONE telephone, but provides a usable 64-character set and a hard-copy history of both input and output. The act of inputting and the ease of comprehension of the output are enormously simpler. The increase in complexity and cost yields a disproportionately large increase in overall utility.

After that, a small but important step is that typified by such typewriter terminals as the Model 37 Teletypewriter [1] and the General Electric TermiNet-300 (Registered Trade Mark of the General Electric Company) [2]; they provide the full ASCII [3] character set (including 95 graphics) and relatively good quality hard-copy at a somewhat higher speed (15 and 30 characters/second respectively). Terminals in this class typically cost several times more than a Model 33 (although their cost is not much more than that of the Model 35 Teletypewriter [1], which may be considered a rugged version of the Model 33). The larger character set (as well as other features) provided on these terminals permits important additional applications. These include the use of the more modern programming languages and the general area of document preparation and maintenance. The step up from a terminal in the Model 33 class to one in the Model 37 class involves little or no increase in the complexity of use and no loss of capability. The additional general utility seems worth the additional cost.

The next step is less an increase in hardware complexity than a trading of mechanical for electronic complexity. These are terminals in which the hard-copy printer is replaced by a character CRT display. They usually achieve an increased output rate some four to ten times higher (when restricted by voice-grade transmission) at a moderate increase in cost. Frequently, some form of local editing is provided for text locally stored in the display buffer. The price of this additional capability, however, is the loss of the hard-copy capability. The net change in terminal utility is not obvious. The truly remote user not located near the computer center line printer has no way of getting hard-copy. The ability to display computer output at a somewhat faster rate is not a complete substitute for a hard-copy history of earlier work. One of the more serious losses resulting from the lack of hard-copy is the loss of the ability to work off-line.

The local editing provided by terminals in this class is basically character-string insertion and deletion, and is inferior to most software text editors available today. It seems that the lack of hard-copy is not balanced by the somewhat higher speed and local editing, and that the net terminal utility is somewhat less than that of those in the Model 37 class.

Next are terminals having true graphical capability (able to draw more or less arbitrary lines). The simplest type of graphical terminal uses a storage tube display device obviating the need for other picture storage. After these come the graphical terminals capable of fast changing pictures, local picture modification, graphical input, and other sophistication. These terminals often contain considerable computing power. Terminals providing true graphical ability undoubtedly open the door to a potentially large increase in terminal utility. A graphical output ability alone can, if used, balance the loss of hard-copy. The cost of such terminals compared to those in the Model 37 class ranges from around twice to out-of-sight.

COMPLEXITY AND SUPPORT

The complexity of the hardware and software interface between the terminal and the computer can affect the terminal utility by affecting the degree of support by the computer system. The TOUCH-TONE telephone need for audio computer output is expensive to satisfy in any general way. On the other hand, hardware support at the computer system for typewriter-like terminals is not overly expensive. Because of generally more sophisticated applications the software support of graphical terminals takes longer to accumulate to useful proportions; and often because of the desire to use faster transmission facilities, the hardware support is more expensive. Currently, the cost and extent of terminal support seems optimal in the area of typewriter-like terminals.

CONCLUSIONS

The significantly higher cost of true graphic terminals can be expected to limit their wide-spread use for some time. Today, the optimum balance between terminal cost and terminal utility seems to point to the continued wide-spread use of typewriter-like terminals, with the trend upward toward those in the Model 37 class. Of course, the cost of the electronic portion of terminals can be expected to continue to decrease. Non-hard-copy, character CRT terminals are probably not the best direction for future evolution. Features such as local editing are better left to central site software and not frozen into large scale integrated hardware.

The ideal terminal for wide-spread general use and utility would be one providing both hard-copy and at least a graphical output capability. The compact fusion of a Model-37-like terminal with a storage tube display for a total cost comparable to today's Model 37 class terminal would be a worthwhile product.

REFERENCES

[1] Model 37 Teletypewriter Stations for DATA-PHONE (Registered Trade Mark of the Bell System) Service, Bell System Data Communications Technical Reference, American Telephone and Telegraph Company, September, 1968. Similar documents are available for the Model 33 and the 35.

[2] Programmers' Manual for the TermiNet-300 Printer, No. GEK-15002, General Electric Company, September, 1969.

[3] American National Standard X3.4-1968, American National Standards Institute, October, 1968.

HIGH BIT DENSITY, CLOSED FLUX-PATH MAGNETIC THIN FILM MEMORY — FINE STRIPED MEMORY

S. Oshima, T. Kobayashi, K. Amano, T. Kamibayashi,
A. Okada, Y. Komazawa and K. Komuro
Research and Development Laboratory,
Kokusai Denshin Denwa Co., Ltd. Tokyo, Japan
(Japan's Overseas Radio and Cable System)

ABSTRACT

In this paper, new type of high bit density, closed flux-path magnetic thin film memory named Fine Striped Memory is proposed and its fabrication techniques, structural details and memory characteristics are described.

Fine Striped Memory is batch-fabricated applying the techniques of vacuum evaporation and photo-etching of Permalloy and copper films, and electroplating of Permalloy film. Width of strip can be made 50 microns or less, and with the use of efficient ferrite flux keeper, bit density of 5, 000 bit/cm^2 is already achieved and 10, 000 bit/cm^2 will be possible in the near future.

Experimental results obtained so far showed very satisfactory memory performance under relatively low digit and word drive current. Although there are several problems to be solved before its practical application, Fine Striped Memory appears to be one of the most promising memories as high-speed, large capacity and potentially low cost internal memory in the future.

INTRODUCTION

In the development and progress of digital computer, internal memory has played a very important role, and various attempts have been made to increase the operating-speed and storage-capacity of internal memory, and to lower the cost of memory devices. Although ferrite core memories have been the best established and most widely used memory element, various types of memories using magnetic thin film and semiconductor integrated circuit have been proposed and developed recently.

In Research and Development Laboratory of Kokusai Denshin Denwa Co., Ltd., studies have been done about the application of magnetic thin films to high-speed memory devices for a long time, and recently successful results have been obtained in fabricating a high density magnetic thin film memory with closed flux-path, applying the techniques of vacuum evaporation and photo-etching of Permalloy and copper films, and electroplating of Permalloy film. Bit density of 5, 000 bit/cm^2 is already achieved and 10, 000 bit/cm^2 will be possible in the near future. This type of memory is named as Fine Striped Memory.

In this paper, the process of development, the fabrication techniques, the structure and the memory characteristics of Fine Striped Memory are described in detail, and remaining problems are indicated.

DEVELOPMENT OF FINE STRIPED MEMORY

Although it was recognized that magnetic thin film has many excellent properties as high-speed, large capacity memory element, and has been studied for a long time, its practical application to memory device is rather delayed. The main reasons are as follows.

1) Output voltage is considerably low.
2) As the size of memory element becomes small, the memory characteristics are affected strongly by the demagnetizing force and domain wall creeping phenomenon is apt to occur.
3) It is very difficult to fabricate and reproduce the magnetic films of uniform and stable characteristics.

These difficulties were overcome in the case of plated wire memory, in which Permalloy thin film of about 1 micron ~ 7, 000Å thickness is electroplated on phosphor bronze wire of 0.1 ~ 0.2 mm in diameter.[1]

As the easy axis of magnetization is circumferential, magnetic flux forms complete closed-path, and consequently plated wire memory is not affected from external interfering field and also produces high output voltage. In Japan, plated wire memory was first developed in a practical form at Research and Development Laboratory of Kokusai Denshin Denwa Co., Ltd. in 1963.[2, 3] Since then, the development of plated wire memories has progressed rapidly and they are already in the stage of practical application to high-speed internal memory.

Although plated wire memory has many features as described above, there are several demerits as follows.

1) The batch-fabrication technique, which is one of the most important merits of magnetic thin film,

cannot be applied.

2) High bit density cannot be achieved.

3) It is difficult to attach efficient flux-keeper.

K. D. D. Research and Development Laboratory has continued the research on a new type of planar thin film memory to get rid of the above-mentioned demerits of plated wire memory, and recently has succeeded in fabricating high bit density memory with closed flux-path. (Fine Striped Memory).[4, 5] The detail of this memory will be described in the following sections.

FABRICATION PROCESS AND STRUCTURE OF FINE STRIPED MEMORY

The fabrication process of Fine Striped Memory is illustrated in Figure 1. Permalloy film, which is composed of 81 per cent Nickel, 16 per cent iron and 3 per cent Cobalt, and whose thickness is about 1,500Å ~ 2,000Å, is vacuum evaporated on the whole area of glass substrate in the appropriate uniform magnetic field, and after that, copper film whose thickness is about 3,000Å ~ 1 micron is vacuum-evaporated on the Permalloy film.

Then photo-resist (KPR) coating is applied over the evaporated film and exposed to intense light after the film is covered with stripe-shaped mask. The hatched area in Figure 1 (a) shows exposed parts. After photographic development, unexposed photo-resist is removed. Fixing the photo-resist which is not removed in development, the substrate is dipped in etchant, and the exposed part of copper and Permalloy are etched, then fine stripes are formed as shown in Figure 1 (b).

After stripping residual photo-resist and cleaning up, Permalloy thin film (81 Nickel, 19 iron) is electroplated on the copper film with the thickness of about 2,000Å in uniform magnetic field whose direction is orthogonal to strip length. As a result, a number of fine strip lines with closed flux path in circumferential direction are batch-fabricated on the glass substrate as shown in Figure 1 (c). The easy direction of magnetization is also circumferential.

The flow chart of the fabrication process is shown in Figure 2. Selection and cleaning of glass substrate are necessary in order to avoid the distortion, scratches and dusts on the surface. Special techniques are required in vacuum evaporation, photo-etching and electroplating of Permalloy film. For example, the evaporation of Permalloy film whose thickness is about 300Å over copper film is effective to improve magnetic characteristics of electroplated Permalloy film. Figure 3 shows enlarged top view and Figure 4 shows enlarged side view of cross section of fine stripes. Figure 5 shows one example of whole view of this memory.

These memory strips are sandwiched by word lines with ferrite flux keeper backed and return lines, which makes right angle with strip lines. The central copper layer of the strip is used as digit and sense line. The structural details are shown in Figure 6. Figure 7 shows one example of the enlarged cross section of grooved ferrite flux keeper. The distance and gap of groove are 100 microns, and depth of groove is about 500 microns.

In order to store the binary information to this memory, word drive current pulse is applied to the word lines and at the same time, digit current pulse whose polarity corresponds to the binary information to be stored is applied to the digit line. To read out the stored information, word drive pulse is applied to word line and rotates the direction of magnetization of Permalloy film, then output signal whose polarity corresponds to the stored binary information is induced in the digit line.

Because digit line is surrounded by two kinds of Permalloy film, i. e. evaporated and electroplated film, it is easy to vary the coercive forces of these two films. If coercive force of one film is higher than that of other film, the film of higher coercive force can be used as medium to store the information and film of lower coercive force can be used as medium to sense the stored information. Therefore, non-destructive read-out operation may be realized similar to the case of coupled-film memory.

Fine Striped Memory has following features.

1) As the digit line is surrounded by Permalloy films and magnetic flux-path is closed, the digit current is remarkably reduced. The closed flux structure in memory strip keeps off the ill-effect on memory characteristics due to demagnetization which occurs when the film thickness is increased. The output voltage decreases proportionally as the distance between word lines reduces. But this is compensated by the increase of memory film thickness. Moreover, as no external field affects the information flux, little noise is produced on the output. Interference from the adjacent digit lines is dismissed. This assures that digit lines are very closely arranged and high density fabrication is achieved.

2) Since this memory strip has flat surface on its top, the flat type ferrite flux keeper which is grooved in parallel, as shown in Figure 6, can be completely contacted to the memory cell. This assures that the word drive flux also passes through the almost closed-flux path, and consequently, the density of memory cell increases and the word drive current decreases. Furthermore, as the drive lines are laid on the bottom of the grooves, the capacitive coupling between drive and sense line is reduced. Thus, the higher signal to noise ratio in output signal is obtained.

3) Photo-etching assures very fine fabrication and has advantage in the production of very high density memory cells. This memory needs only one time to align the pattern mask in the process of fabrication and

such a batch fabrication promises very low cost in production.

EXPERIMENTAL RESULTS

Coercive forces of the evaporated or electroplated magnetic thin films are measured by the equipment using Kerr magneto-optic effect. Examples of B-H curves of electroplated thin films are shown in Figure 8 for easy and hard directions of magnetization.

Figure 9 shows the relation between word drive current Iw versus output voltage Vo when digit current Id is 10 mA with pulse train format which was used to measure the characteristics. The cross section of ferrite keeper and word line is also shown. The dimensions of memory strip are as follows.

1)	width of memory strip	50 microns
2)	thickness of evaporated Permalloy film	1, 500Å
3)	thickness of copper film	3, 000Å
4)	thickness of electroplated Permalloy film	2, 000Å.

The output voltages are measured at two cross points / bit and rise time of word drive current is 20 nS. The dotted line shows the output voltage when word drive current is applied immediately after the information is stored, and the solid line shows the output voltage when 50 digit disturbance pulses of opposite polarity are applied before the information is read out. There is not much difference between these two curves.

Interferences from adjacent word lines are also checked and they are sufficiently suppressed with grooved ferrite flux keeper. Almost same performances are obtained in case that the width of the memory strip is about 30 microns.

Figure 10 shows the relation between digit current Id versus output voltage Vo when word drive current Iw is 300 mA. Dimension of memory strip and pulse train format are same as in Figure 9. This shows wide operating region and stable memory operation are achieved.

Typical output waveforms with 50 digit disturbance pulses are shown in Figure 11. Figure 12 shows the comparison of typical output waveform of Fine Striped Memory with that of plated wire memory. It is evident that clear output waveform is obtained at low word drive current in Fine Striped Memory.

In Figure 13 another type of word line is employed, which is surrounded partially by Permalloy film as shown in the cross section in the figure. The relation between the thickness of spacer and output voltage is shown. The solid line corresponds to the output voltage when no digit disturbance pulse is applied and the dashed line corresponds when 50 digit disturbance pulses are applied. The dotted line corresponds when 50 digit disturbance pulses and adjacent word drive current are applied. If the thickness of spacer is below 13 microns, output voltage is almost constant and flux of word drive current concentrates on memory strip.

Uniformity of output voltage on the substrate plane is one of the most important factors. Figure 14 (b) shows one example of output voltages of various parts which correspond to the points on the plane in Figure 14 (a). Though considerable variance is observed in the peripheral region of the plane, almost uniform output voltages are obtained from the major part of the plane.

FURTHER PROBLEMS

The width of memory strip used in the experiment is 50 microns. However, it is confirmed that the width of memory strip and the distance between memory strips can be decreased to 20 microns. The distance between word lines can be reduced to 200 microns. Therefore memory density of about 10, 000 bits/cm^2 can be realized in the future.

In the experiment in the proceeding section, the output voltages are measured in destructive read-out mode. However, non-destructive read-out operation will be feasible as the magnetic characteristics of thin film will be improved, and this will be the important subject for a further study.

Some unsolved problems remain when the practical use of Fine Striped Memory is considered. Because the distance between digit lines is very narrow, it is rather difficult to interconnect the Fine Striped Memory to another Fine Striped Memory or peripheral circuits. Wire bonding techniques which are utilized in connecting terminals of IC chip to output terminal lead should be used in the interconnection of Fine Striped Memory.

Other important factors are stability and uniformity of magnetic characteristics of Fine Striped Memory plane. Although long range stability and reliability of Fine Striped Memory are not yet confirmed, stable characteristics are expected with the analogy to the case of plated wire memory.

Furthermore, integration of peripheral circuits, i. e. word drive circuits, digit drive circuit and sense amplifier, is absolutely necessary and a considerable effort must be spent for this field.

CONCLUSION

In this paper, new type of high bit density closed flux-path magnetic thin film memory named Fine Striped Memory is proposed and its fabrication process, structural details and memory characteristics are described.

Fine Striped Memory is batch-fabricated applying the techniques of vacuum evaporation and photo-etching of Permalloy and copper films, and electroplating of Permalloy film.

Width of memory strip can be made 50 microns or less, and with a use of efficient flux keeper, bit density of 5,000 bit/cm^2 is already achieved and 10,000 bit/cm^2 will be possible in the near future. Experimental results obtained so far showed satisfactory memory performances under relatively low digit and word drive current. Although there are several problems to be solved before the practical application, the authors believe that Fine Striped Memory is one of the most promising memory as high-speed large capacity and potentially low cost internal memory device in the future.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Shinkawa, Director of Research and Development Laboratory of Kokusai Denshin Denwa Co., Ltd., for his continuing guidance and encouragement.

REFERENCES

1. T. R. Long "Electrode posited Memory Elements for a Non-destructive Memory," J. Appl. Phys. Supplemented to Vol. 31, pp. 1235, May 1960.

2. S. Oshima and T. Kamibayashi "Wire Memory Matrix," Jour. of I. E. C. E., Japan Vol. 47, No. 7 (July 1963) p. 943.

3. S. Oshima, K. Futami and T. Kamibayashi "The Plated-wire Memory Matrix," Proc. 1964. INTERMAG Conf. 5-1-1.

4. S. Oshima "Recent Developments of Magnetic Film Memories in Japan," Proc. 1969. INTERMAG Conf. 13.1.

5. S. Oshima, T. Kobayashi, T. Kamibayashi, A. Okada, Y. Komazawa and K. Komuro "Fine Striped Memory," Papers of Technical Group on Electronic Computer EC-69-7 (May 1969).

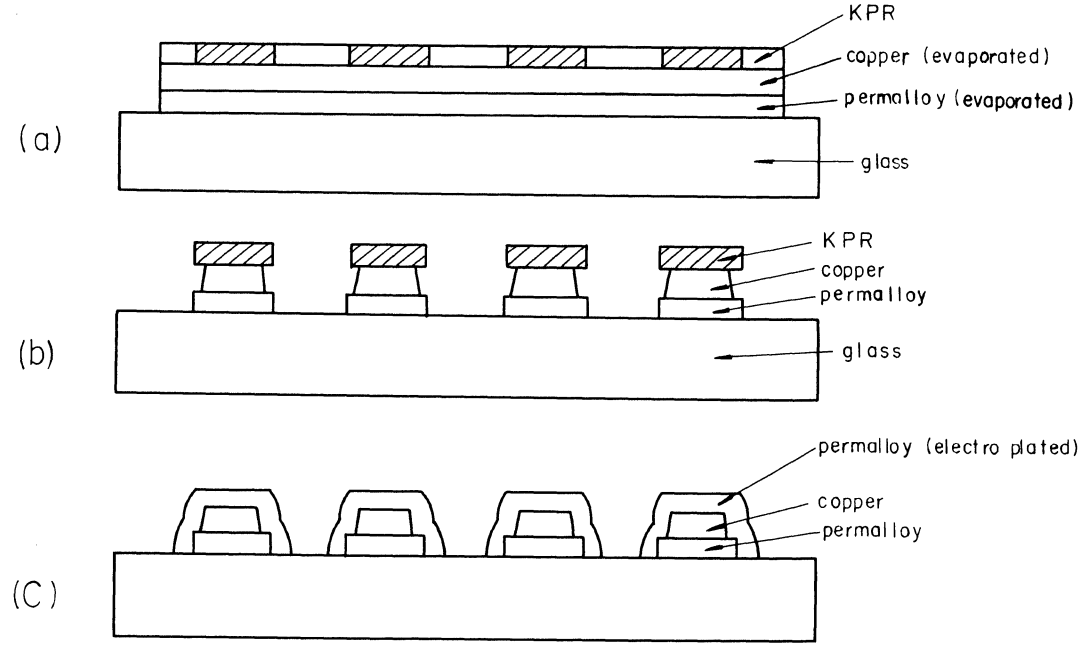

KPR
copper (evaporated)
permalloy (evaporated)
(a)
glass
KPR
copper
permalloy
(b)
glass
permalloy (electro plated)
copper
permalloy
(C)

Fig. 1 Fabrication process of Fine Striped Memory.

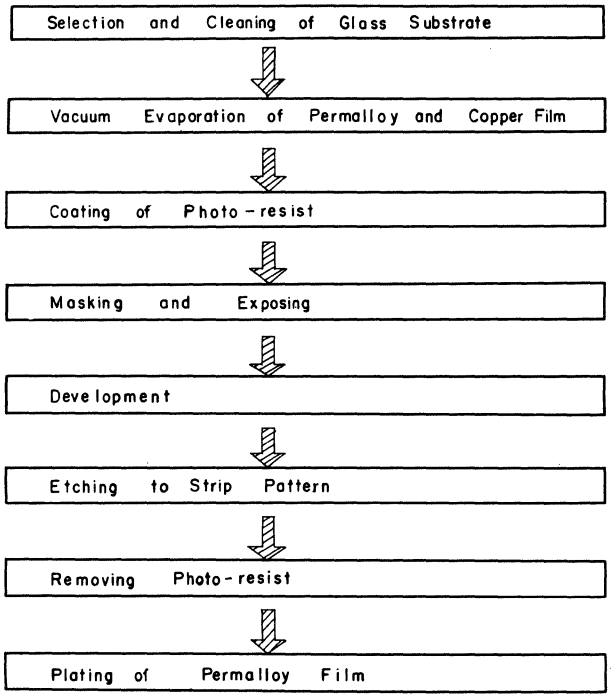

Selection and Cleaning of Glass Substrate
Vacuum Evaporation of Permalloy and Copper Film
Coating of Photo-resist
Masking and Exposing
Development
Etching to Strip Pattern
Removing Photo-resist
Plating of Permalloy Film

Fig. 2 Flow chart of fabrication process.

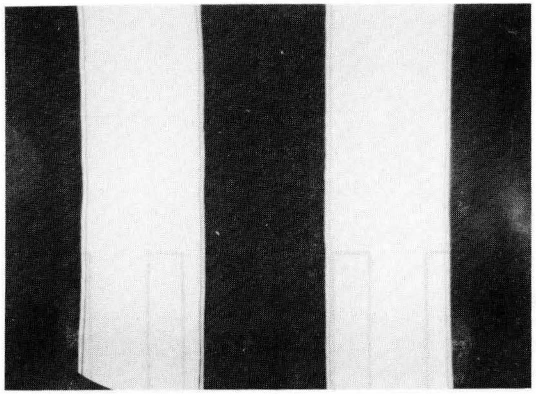


Fig . 3 Enlarged top view of memory strip
white line — memory strip (width 50μ)
black line — glass substrate (width 50μ)

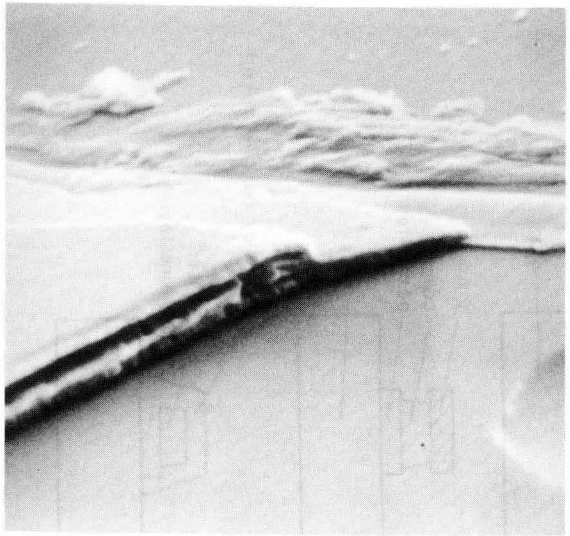


Fig . 4 Enlarged view of cross section of memory strip
thickness of evaporated Permalloy — 1500 Å
thickness of copper — 3000 Å
thickness of electroplated Permalloy — 2000 Å
width of overlapped part of Permalloy films — 3μ

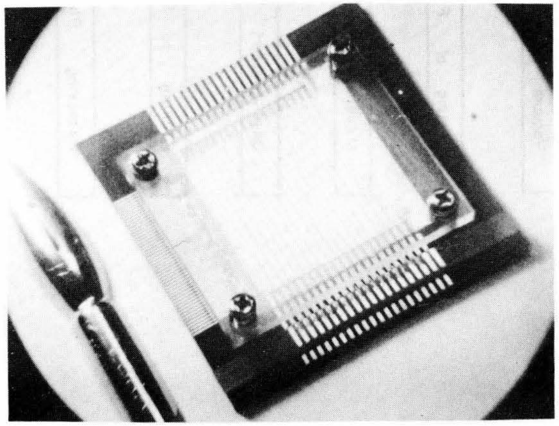


Fig. 5 One example of whole view of Fine Striped Memory
(ferrite keeper removed)
digit line pitch — 400μ
word line pitch — 500μ

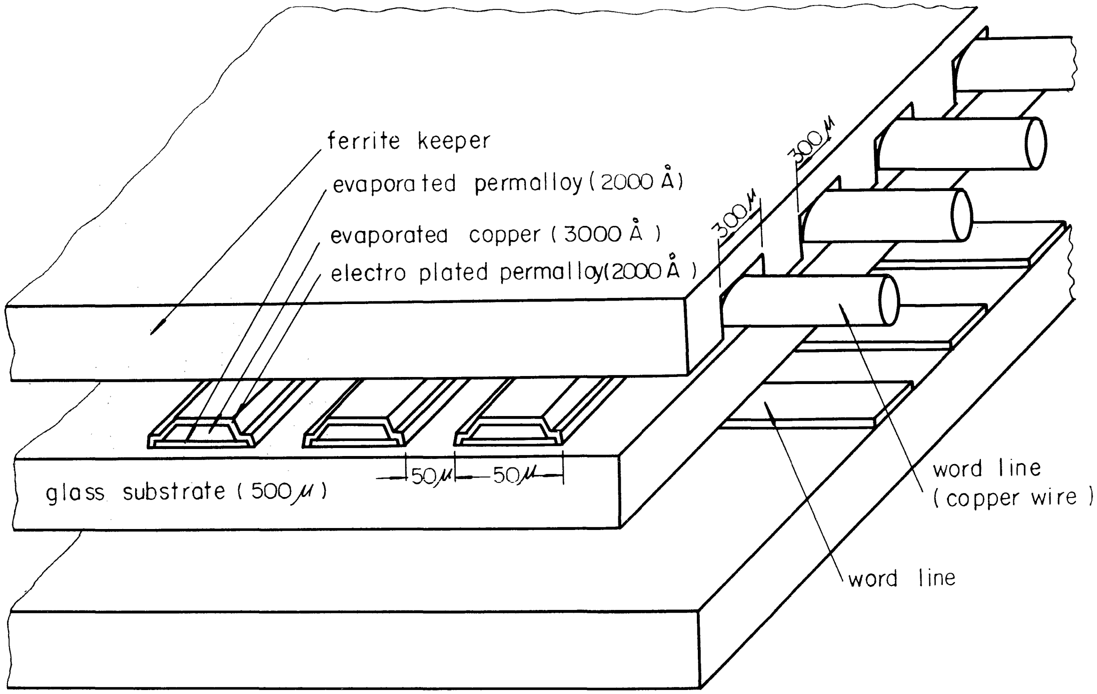

ferrite keeper
evaporated permalloy (2000 Å)
evaporated copper (3000 Å)
electro plated permalloy (2000 Å)
300μ
300μ
glass substrate (500μ)
50μ
50μ
word line
(copper wire)
word line

Fig. 6 Structure of Fine Striped Memory.

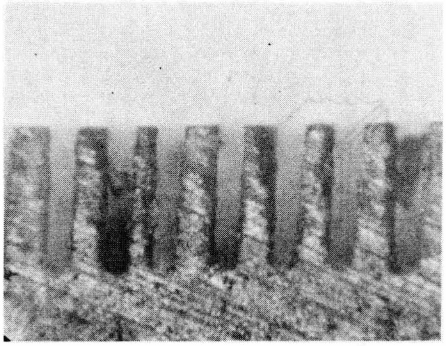


Fig. 7 Enlarged cross section of grooved ferrite keeper.
gap and width of groove — 100μ
depth of groove — 500μ

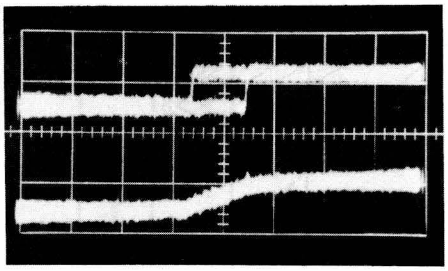


Fig. 8 B-H curves of electroplated Permalloy film.
upper curve - easy direction of magnetization
lower curve - hard direction of magnetization
thickness of Permalloy film — 2000 Å
horizontal scale — 5 Oe/div.

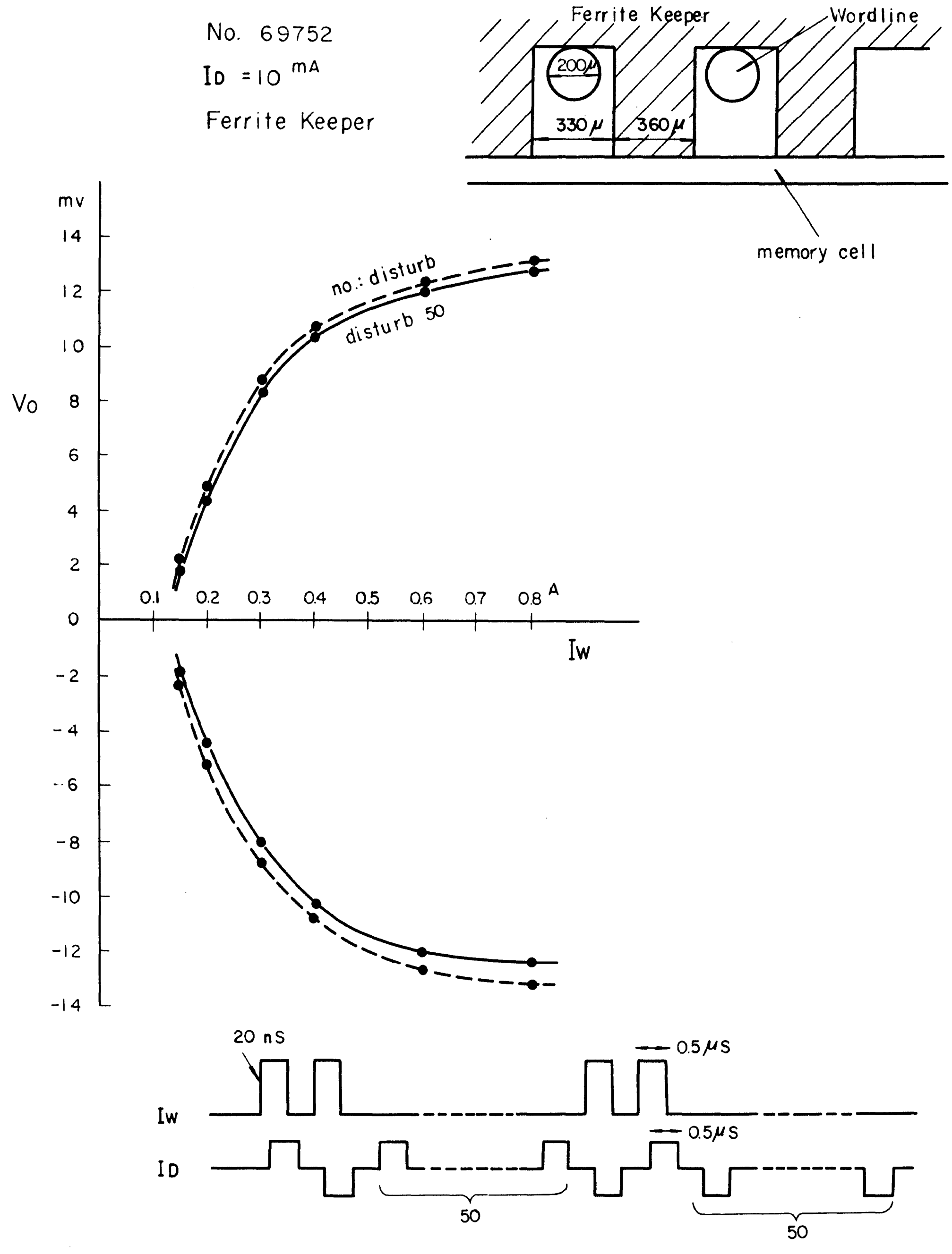


Fig. 9 Word drive current versus output voltage.

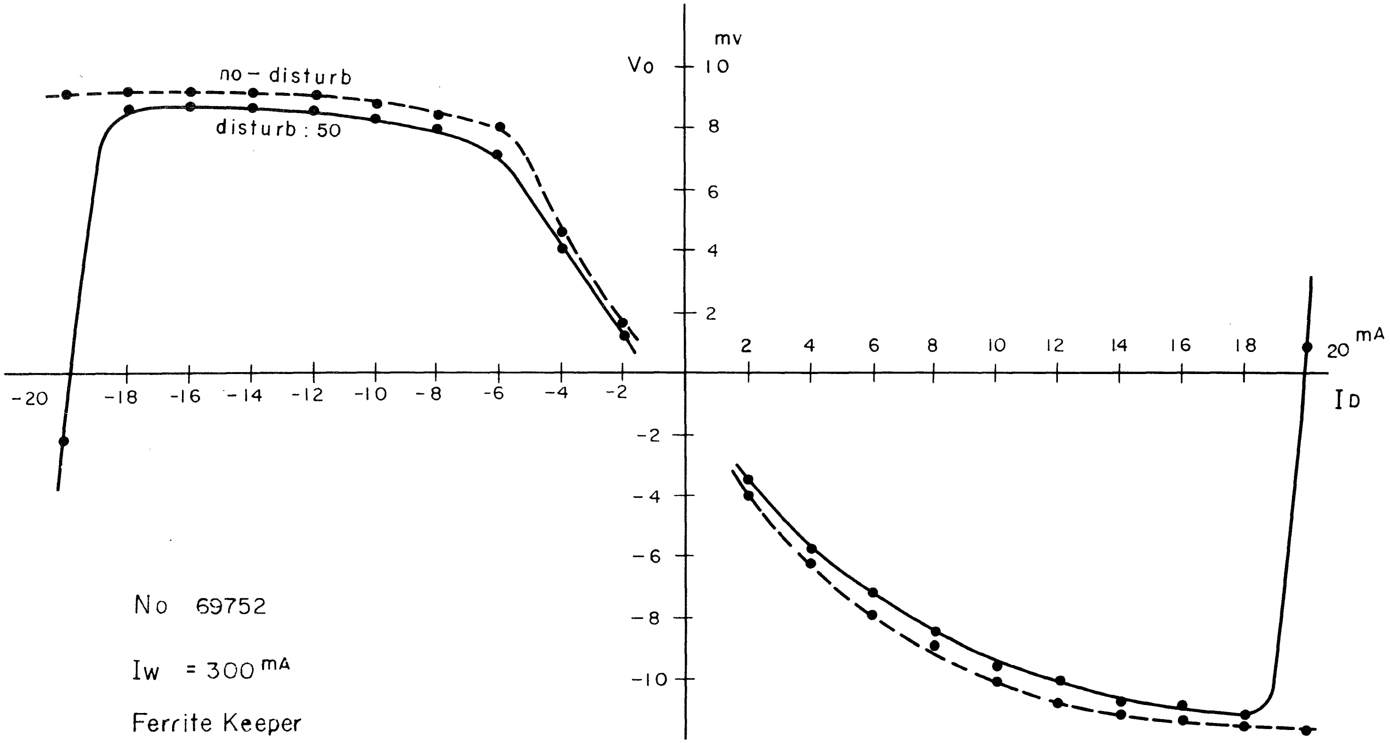


Fig. 10 Digit current versus output voltage.

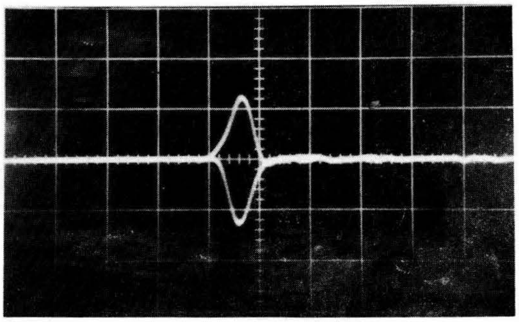


Fig. 11 Output waveform of 2 cross points / bit.

width of memory strip — 50 μ
word line pitch — 700 μ
Iw (word current) — 0.3 A (rise time 20 nS)
ID (digit current) — 10 mA
number of disturbance — 50 pulses
vertical scale — 5 mV/div.
horizontal scale — 20 nS/div.

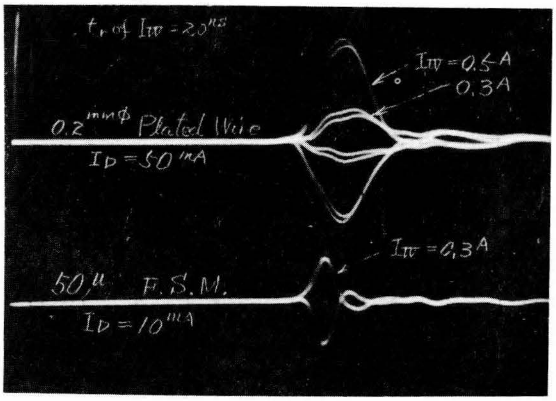


Fig. 12 Output waveforms of plated wire memory and Fine Striped Memory.

upper - plated wire memory without keeper
lower - Fine Striped Memory with keeper

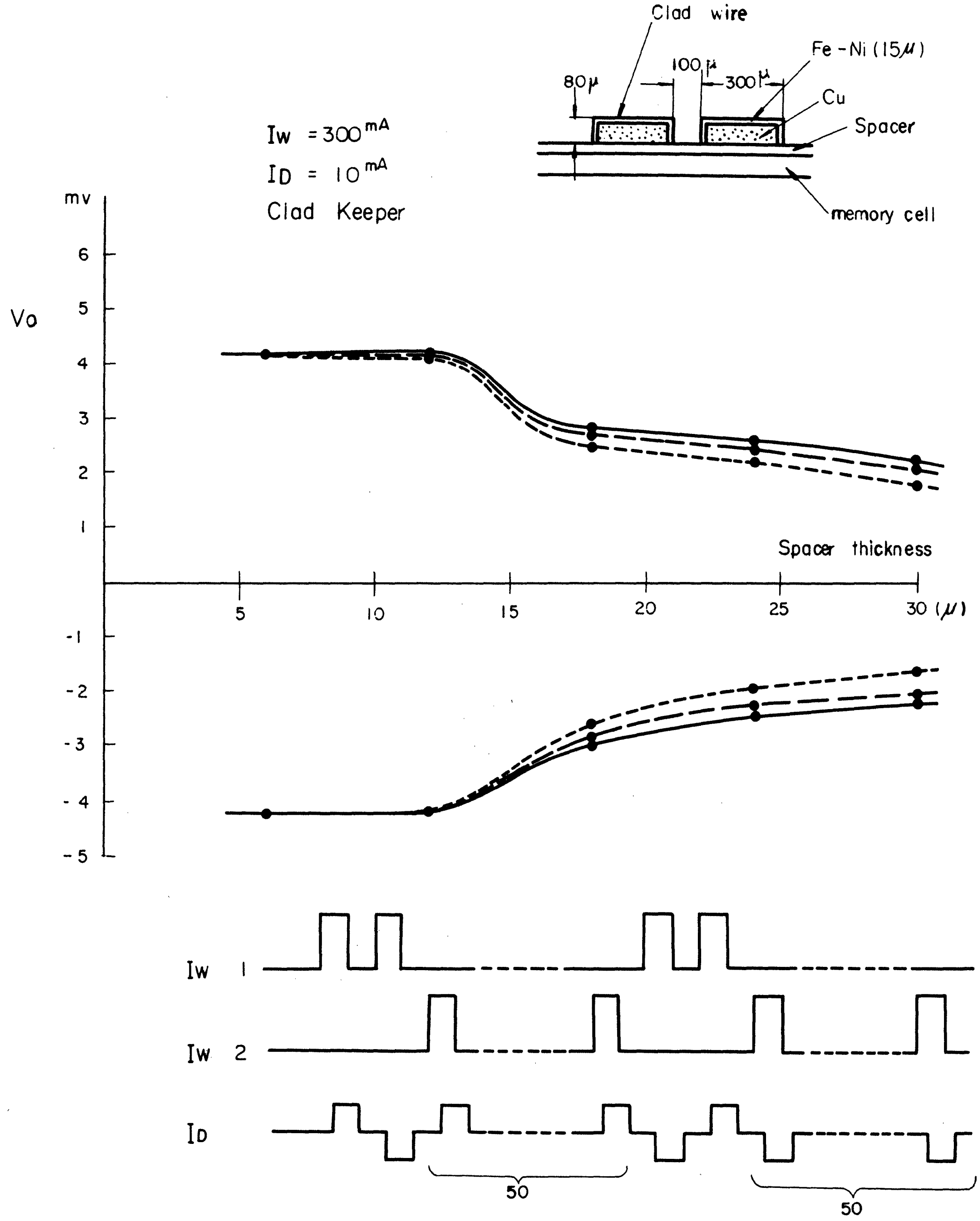
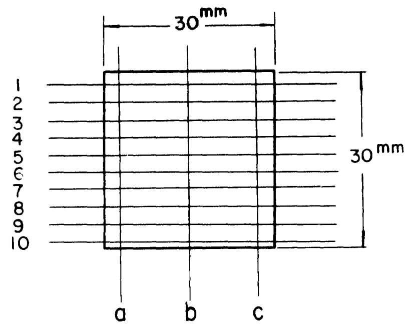


Fig. 13 Output voltage versus spacer thickness (clad wire is used as word line)



(a)

	a		b		c	
	+	—	+	—	+	—
1			8.0	8.0		
2	9.5	9.0	8.0	8.5	5.0	6.5
3	11.0	10.5	8.0	9.5	7.0	7.0
4	9.5	8.0	9.0	10.0	7.0	7.5
5	10.0	10.0	7.0	9.0	6.5	6.0
6	7.5	9.0	9.0	9.0	7.5	8.0
7	9.5	10.0	9.0	11.0	6.5	8.0
8	9.0	13.0	9.0	11.0	8.0	9.0
9	11.5	10.0	11.0	12.0	8.0	10.0
10	10.0	12.0	11.0	11.0	9.5	11.0

(b) Output voltage (mV)

Fig. 14 One example of uniformity of output voltage on the plane.

Output signals from the cross points of lines in Fig.14(a) correspond to output voltages in Fig.14(b) respectively.

A PROPOSED FAST-ACCESS LARGE-CAPACITY FERROELECTRIC MEMORY

D. W. Chapman, Systems Development Division, IBM Corp., San Jose, California

ABSTRACT

Recent work on polycrystalline ferroelectric materials suggests it may be feasible to prepare photoconductor-ferroelectric reversible memory chips for use in multichannel CRT-type flying-spot stores. The storage chips would operate as light-controlled voltage dividers, with the large dark-to-light resistance ratio of the photoconductor providing auxiliary thresholding to compensate for the lack of a true coercive force in the ferroelectric. The multichannel nature of the store would compensate for the slow response time of the photoconductor. A store driven by one CRT could have a capacity between 10^7 and 10^8 bits and a random reference cycle of 25 microseconds to blocks of 240 bits. The data rate could be increased by operating several stores in parallel, and it should be possible to operate ten or more stores in a single file facility sharing common electronics to achieve $\sim 10^9$ bits capacity at a favorable cost per bit.

I. INTRODUCTION

On-line auxiliary storage in present computer systems is usually implemented by magnetic-disk-file technology. Progress in this technology has permitted direct access to increasing storage capacities ($>10^9$ bits) at rapidly decreasing cost (< 0.1¢ per bit). Its electromechanical nature, however, has prevented reduction of access time below the 10 to 100-millisecond range, and requires synchronous operation with semirandom access to relatively large blocks of data.

Conversely, magnetic core and most other solid-state technologies used in the main internal memories of present systems are asynchronous, and offer completely random access to blocks of data as small as 8 bits. Main memory technology has progressed primarily in the direction of faster access and cycle time, with storage capacity being limited usually to less than 10^7 bits because cost per bit tended to increase with increasing capacity.

The need for auxiliary on-line storage to achieve adequate system capacity at acceptable cost, together with the synchronous operation and slow access rate of the auxiliary storage technology, has resulted in poor throughput and efficiency. Future development of very large-capacity, fast, asynchronous, random-access memories at moderate to low cost per bit could substantially increase throughput and efficiency in present computer applications, and would accelerate the trend toward greater on-line use of computers.

Laboratory tests conducted a number of years ago indicated that capacities in excess of 23 million bits, with random access in the low microsecond region, were feasible and inexpensive in read-only technology using CRT-type flying-spot optical systems combined with silver-halide photographic storage media.[1]

Figure 1 illustrates the type of read-only flying-spot store cited above. Any addressed spot on the CRT raster is simultaneously imaged by the lenses in the objective lens array onto corresponding spots on each chip (film) in the memory panel. In each channel, depending on whether the addressed film spot is transparent or opaque, light is or is not transmitted through a condenser lens into a photomultiplier and generates a one or zero output signal.

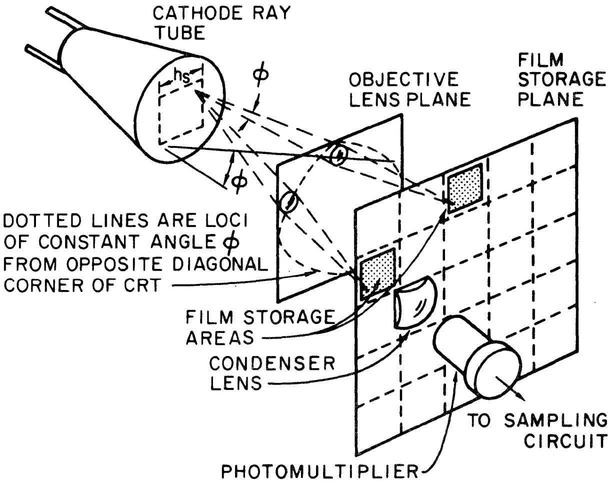


Fig. 1 Form of the flying-spot store (after Hoover's Fig. 1 in Ref. 14).

Each film chip has a storage capacity equal to the number of spots in the CRT raster, and the total capacity of the store is the product of the number of objective lenses (channels) and the number of spots on a single chip. The data rate is the number of data channels divided by the cycle time, and the "word size" in bits equals the number of data channels. A few of the channels are used for positioning the CRT's electron beam with the aid of a closed-loop servo system.

If the nonreversible photographic storage medium used in the above type system could be replaced by a reversible storage medium, and if many such stores were to be incorporated into one facility sharing common electronics, then inexpensive, very large-capacity, read-write storage systems would be feasible with access times being microseconds rather than milliseconds as currently required with magnetic-disk-file technology. Such memories could be used as main memories or auxiliary on-line memories, depending on the speed and cost/performance requirements of particular system designs and applications.

Recent work in our laboratory and elsewhere on polycrystalline ferroelectric materials[2,3] led to the proposal that it might be feasible to prepare photoconductor-ferroelectric memory chips suitable for use as a reversible memory medium in such stores. Thus, at relatively low cost, we might be able to achieve access times ≤ 25 microseconds and data rates ≥ 10 megabits per second in read-write storage facilities having capacities up to 10^9 bits.

II. THE MEMORY DEVICE

A. General Description and Mode of Operation

The photoconductor-ferroelectric memory device was first described by Schaffert;[4] in more recent years, work on multilayer devices combining a ferroelectric with a photoconductor or semiconductor has been described by Hanlet and Requa,[5] by Sihvonen, et al.,[6] by Hell and Asam,[7] and by Heyman and Heilmeier.[8]

As described by Schaffert and experimentally attempted by Hanlet and Requa, Hell and Asam, and as described here, the device would ideally consist of four layers: (1) a conductive substrate, covered by (2) a ferroelectric layer, covered by (3) a photoconductor layer, covered by (4) a transparent electrode. The device is illustrated in Figure 2; it would operate as a light-controlled voltage divider when an electric field is applied between the transparent electrode and the substrate electrode. See Figure 3 for an equivalent circuit.

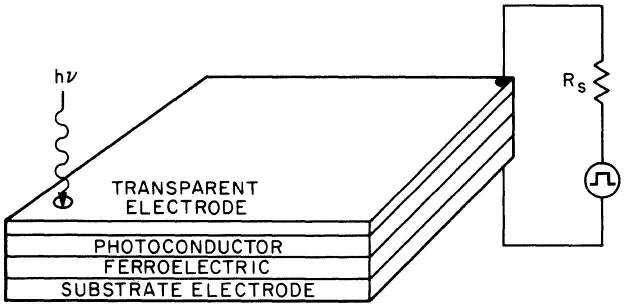


Fig. 2 Proposed memory device.

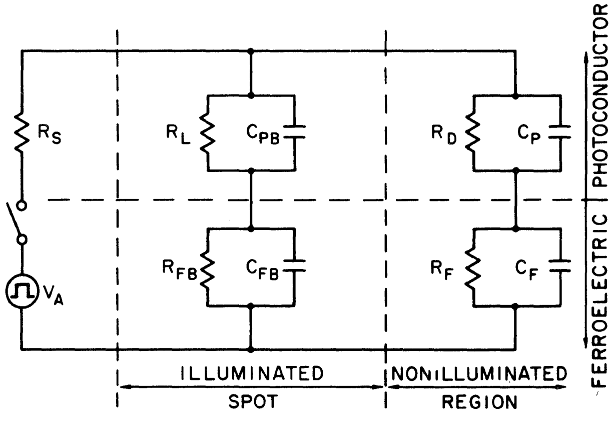


Fig. 3 General equivalent circuit.

If the capacitance of the ferroelectric layer is large compared to that of the photoconductor, and the dark resistance of the photoconductor is large compared to that of the ferroelectric, nearly all of the applied voltage always appears across the photoconductor in the dark, and the ferroelectric is not affected. (Under pulsed conditions, the photoconductor dark resistance would not necessarily have to be larger than the resistance of the ferroelectric.) However, when a sufficiently intense beam of light is incident on a small spot of the photoconductor at the same time that a voltage is applied, the resistance of the photoconductor at that spot can be made small compared with the capacitive reactance of the photoconductor and the resistance of the ferroelectric at that spot. Then -- at that spot only -- the voltage pulse appears across the ferroelectric if the pulse length is long compared with the charging time constant of that small spot on the ferroelectric.

If the applied voltage pulse is large enough and long enough, and if its polarity is chosen properly, it reverses the polarization of the ferroelectric at the location where the light beam is incident on the photoconductor. Thus, this phenomenon can be used to record and destructively read out binary information at light-beam addressed spots on the memory device.

The switching time should be approximately given by the relation[9]

$$T_S \simeq \frac{2 P_R A}{V - V_C} R_L \qquad (1)$$

where P_R, V_C, and V are the remanent spontaneous polarization, coercive "voltage," and voltage applied to the ferroelectric, respectively. A is the area of the illuminated spot and R_L the light resistance of the illuminated region of the photoconductor.

If V is large compared with V_C and R_L is small (intense illumination of the bit region), we have the familiar relation for ferroelectrics that the switching time varies inversely with the applied voltage. For relatively large R_L, the ferroelectric bit is driven by a current source which provides the most reliable operation for V only slightly larger than V_C; however, the switching time is relatively long under these conditions.

In the memory system proposed here, we envisage that readout of a stored bit is destructive, being accomplished by reversing the polarity of the stored bit. When the bit in the memory device switches its polarization, a transient current flows, causing a small voltage pulse to be superimposed on the applied voltage pulse. Detection of this switching transient might then require that the applied pulse also be applied separately to a dummy capacitor, and that a differential amplifier be used to detect the difference between this "dummy pulse" and the pulse across the memory device. An alternate approach might be to integrate the current pulse through the memory device over a complete switching cycle. The current charging and discharging the linear capacitance of the device then integrates out to zero, whereas the integral of the switching transient yields a remanent voltage level across the integrating capacitor. (The much simpler approach of turning the CRT beam on after the applied voltage is established is not attractive because the beam must be on during the latter portion of the access cycle in order for the beam-positioning servo to function.)

The transient switching current is approximately

$$i = \frac{2 P_R A}{T_S}. \tag{2}$$

It flows through R_L, provided R_L is small compared with R_{FB}, and then through the parallel combination of R_S and the impedance of C_p, C_F, and their parallel "leakage" resistances. For reasonably fast switching speeds, however, each leakage resistance is large compared with the capacitive reactance it is shunting; thus, if R_S is small compared with the capacitive reactance of C_p, a stored "1" yields a voltage signal of

$$\Delta V \simeq \frac{2 P_R A}{T_S} R_S \tag{3}$$

where R_S is the appropriate load resistor.

B. Specific Device Objectives

Consider now that the ferroelectric is a layer 4 microns thick, similar to that described by the author.[2] Thus, it has a dielectric constant K_F of about 1000, a remanent spontaneous polarization of about 10 microcoulombs per square centimeter, a remanent coercive field (for 2 microsecond pulses) of 20 kilovolts per centimeter or less, a ratio of activation field to remanent coercive field (α/E_{CR}) of about 1.5, and electrical resistivity $\sim 10^{10}$ ohm-centimeters. Figure 4 illustrates the pulse-switching characteristic of a 20-mil spot on such a layer. (Figure 4 is a reproduction of Figure 7 in Reference 2.)

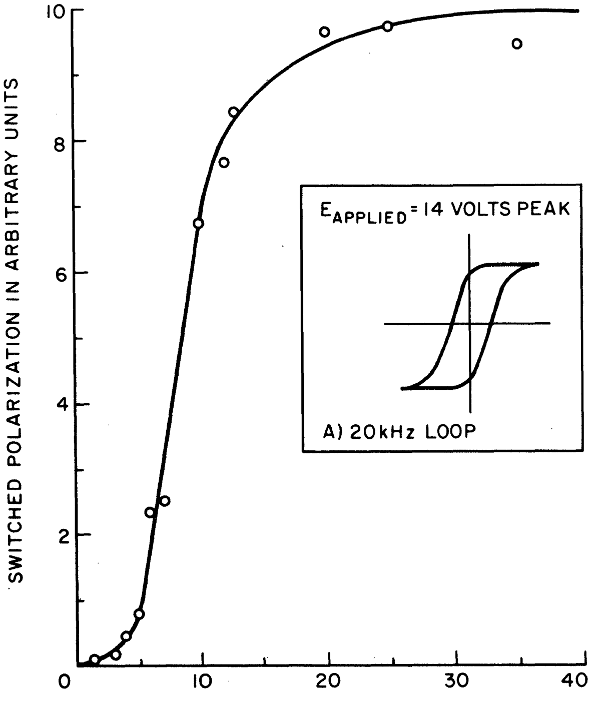


Fig. 4 Remanent polarization switching characteristics of the ferroelectric film for 2-microsecond pulses, starting from negative-saturation remanence.

Let the square memory chip consist of a 0.6 by 0.6-inch conductive substrate, the above layer of ferroelectric material, a 4-micron layer of a photoconductor material having a reasonably large dark resistance and a dielectric constant $K_{PC} \simeq 10$, and a transparent electrode (CdO or SnO_2). Let the chip store 2.5×10^5 bits, each 20 microns in diameter, spaced 30 microns apart (center to center, along each row or column). Furthermore, let us require that the addressed spot switch within 2.5 microseconds.

Because the area of one bit is $\simeq 3 \times 10^{-6}$ cm^2, the linear capacitance of the ferroelectric bit is $\simeq 10^{-12}$ farads, and the charge to be switched $(2 P_R A) \simeq 6 \times 10^{-11}$ coulombs. For switching to occur within 2.5 microseconds, the switching current has an average value $\simeq 25$ microamperes.

Because 20 volts switches the ferroelectric in 2 microseconds, let us assume a voltage pulse of 30 volts is applied across the entire memory device. Then, 2/3 of the applied voltage must appear across the ferroelectric after 0.5 microsecond in order for switching to be completed within a 2.5-microsecond read or write cycle. Hence, the product $R_L C_{FB}$ must be ≤ 0.25 microsecond when the voltage across R_L is ≥ 10 volts (= 1/3 of the applied voltage). Hence,

$$R_L \simeq 2.5 \times 10^5 \text{ ohms} \tag{4}$$

is required.

But 10 volts across 2.5×10^5 ohms = 40 microamperes, which exceeds the average value of the ferroelectric bit's switching current. Thus, $R_L \simeq 1/4$ megohm at 25 kV/cm should be both a necessary and sufficient requirement of the illuminated spot in the photoconductor. This implies a light resistivity

$$\rho_L \simeq 2 \times 10^3 \text{ ohm-cm}. \tag{5}$$

Assuming unity quantum efficiency, uniform illumination through the thickness of the photoconductor, majority carrier conduction and, for reasons discussed in Section III, an irradiance of 1/2 watt per square centimeter at a wavelength of about 5,000 angstroms, the above light resistivity requires a Hall mobility x free-majority-carrier-lifetime product

$$\mu\tau \simeq 5 \times 10^{-7} \text{ cm}^2/\text{volt}. \tag{6}$$

The photocurrent should decay by a couple orders of magnitude within a few microseconds at our irradiance level (1/2 watt/cm^2 $\simeq 10^{18}$ photons/cm^2-sec.) provided the photoconductor's concentration of traps is $\simeq 10^{17}$/cm^3 or less, and the majority-carrier free lifetime $\simeq 1$ microsecond. The required mobility is then essentially $\mu \geq 1$ cm^2/volt-second.

Sommers[10] indicates that all the above requirements conceivably could be met by properly prepared (i.e., reasonably pure) CdS. For example, Sommers claims $\tau_{maj} \simeq 1 \times 10^{-6}$, $\tau_{min} \sim 10^{-7}$, $\mu = 200$ cm^2/volt-sec., $\rho_D > 10^{10}$ ohm-cm, for undoped CdS. ZnCdS and ZnSe would be promising photoconductors for wavelengths shorter than 5,000Å, and CdSe or GaAs for longer wavelengths. Okimura and Sakai report very promising properties in CdSe evaporated films.[11]

We can now calculate the values of all parameters in the equivalent circuit of the device which is now shown in Figure 5 with approximate values. Optimum R_S for resistive readout is determined for this circuit as follows.

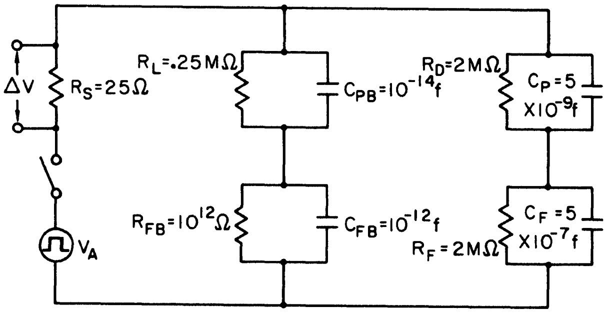


Fig. 5 Specific equivalent circuit.

If the write-drive voltage pulse is to appear across the ferroelectric bit with the R_C time constant of $R_L C_{FB} \simeq$ 0.25 microsecond, then the rise time of the applied pulse, which is determined by the time constant $R_S C_P$, must be equal to or less than 0.25 microsecond.

Thus, we let

$$R_S \simeq 25 \text{ ohms}. \tag{7}$$

The readout signal is therefore

$$\Delta V \simeq \frac{2 P_R A}{T_S} R_S = 0.6 \text{ millivolt}. \tag{8}$$

Because the readout process described here is destructive, a read-rewrite operation must be performed at each spot addressed by the beam, and thus a total beam-dwell time of about 5 microseconds would be required with an actual memory chip.

C. One-to-Zero Ratio and Disturb Sensitivity

As previously mentioned, the majority of the applied voltage would appear across the photoconductor rather than the ferroelectric at the unaddressed bit locations. An undesirable property of nearly all ferroelectrics known to date, however, is that the cumulative effect of many pulses smaller than the coercive force ultimately reverses the polarization of the material. This property is generally referred to as "the lack of a true coercive force."

The lack of a true coercive force means that whenever a write or read operation is performed at any addressed bit, a small amount of polarization at each unaddressed bit is switched. This can generate two problems. One is that the sum of the tiny currents from all the partially switching unaddressed bits causes a readout pulse, even though the addressed bit might be a "0." Thus, although the readout switching transient might be separated from the applied pulse as previously described, there may still be a poor "1" to "0" ratio in readout. An attempt to estimate the worst-case ratio is as follows.

For a field E_F applied to the ferroelectric in the form of a step function, Merz[12] has shown that the maximum value of the switching current transient is given by

$$i_{max} = p\, e^{-\alpha/E_F} \tag{9}$$

where α is an activation field related to the probability of forming domains, and p is a constant. (More accurately, p is constant only for $i = i_{max}$; Drougard[13] has derived a general expression for switching current in which p is shown to be somewhat influenced by the net relative polarization.)

Because only 2/3 of the 30 volts applied to the memory device is applied to the ferroelectric at the addressed bit location, and because the ferroelectric's "remanent coercive voltage" is 8 volts, the amplitude of a "1" readout signal will be proportional to

$$i_{max_1} = p\, e^{-\frac{\alpha}{2.5 E_{C_R}}} . \tag{10}$$

But the maximum field which appears across the ferroelectric at the unaddressed bit locations is

$$\simeq 4 E_{C_R} \frac{K_{PC}}{K_F} ,$$

and thus the total contribution of all the (b-1) unaddressed bits to the readout signal will be proportional to

$$i_{max_0} = (b-1)\, p\, e^{-\frac{\alpha K_F}{4 E_{C_R} K_{PC}}} \tag{11}$$

where b is the total number of bits on a "chip." Hence, the "1" to "0" ratio is approximately given by

$$R_{1/0} \simeq \frac{i_{max_1}}{i_{max_0}} = \frac{e^{\frac{\alpha}{E_{CR}}\left(\frac{K_F}{4 K_{PC}} - \frac{1}{2.5}\right)}}{b-1}$$

$$\simeq \frac{e^{25 \frac{\alpha}{E_{CR}}}}{b-1} \simeq \frac{10^{16}}{b-1} \simeq 4 \times 10^{10} . \tag{12}$$

The read signal generated by the partial switching of the unaddressed bits should therefore be negligible.

The second problem which could be caused by the lack of a true coercive force is that, after many operations, the cumulative effect of the disturb pulses will destroy the information stored in the ferroelectric layer. Thus, after a certain number of operations, it will be necessary to regenerate all the information stored in the memory.

The number of operations possible before regeneration is required can be estimated with the aid of Equations (10) and (11), as follows. The switching transients from each bit location are roughly triangular. (Those from the unaddressed spots tend to be "ramp" shaped -- will never actually reach a value as large as i_{max} before the applied field is shut off -- whereas those from the addressed spot are nearly symmetrical.) Hence, the relative amounts of polarization switched are essentially proportional to i_{max}, and the amount of polarization switched each cycle at an unaddressed spot relative to that at the addressed spot is roughly given by

$$\frac{i_{max_0}/(b-1)}{i_{max_1}} \simeq e^{-\frac{\alpha}{E_{CR}}\left(\frac{K_F}{4\,K_{PC}} - \frac{1}{2.5}\right)} \quad . \tag{13}$$

If the polarization of an unaddressed spot were to be reduced by 20%, this would reduce its future signal output (proportional to 2 P_R) by 10% if it were to be later interrogated as a coded "1." Assuming that this 10% is the limit of safe degradation of the stored information, the number of operations possible before regeneration is approximately the number of operations N given by

$$\frac{N\, i_{max_0}/(b-1)}{i_{max_1}} = 1/5 \tag{14}$$

and we have

$$N \simeq 1/5\, e^{\frac{\alpha}{E_{CR}}\left(\frac{K_F}{4\,K_{PC}} - \frac{1}{2.5}\right)} \simeq \frac{e^{25\frac{\alpha}{E_{CR}}}}{5}$$

$$\simeq 2 \times 10^{15} \quad . \tag{15}$$

Hence, practically speaking, regeneration of the stored information should never be required.

III. PROPOSED STORE CONFIGURATION AND ESTIMATE OF PERFORMANCE

Consider a multichannel flying-spot store, such as that illustrated in Figure 1, and described by Hoover.[1,14] As shown in the Appendix, the store design can be based on the following components:

(1) A slightly scaled-up magnetic-deflection version of the CRT described by Cooper[15] to provide $\simeq 2.5 \times 10^5$ ten-mil spots in a 512 by 512 array on a P22B phosphor. Let the spots be spaced 14 mils apart (center-to-center) on a 7 by 7-inch raster area. Use a beam current of 50 to 100 microamperes, and 20 kV accelerating potential.

(2) Two hundred and fifty-six 3.5-inch focal length, f/8, triplet-type lenses, corrected for 15° half-field angle. These are to be mounted in a 16 by 16 array with center-to-center spacings of about 1.5 times their effective aperture to form a square lens panel measuring about 1 foot by 1 foot. This yields 240 data channels and 16 servo channels for beam positioning. The lens panel, positioned about 46 inches from the CRT faceplate, images the CRT spots on the memory device plane with a spot-diameter reduction of 12. Hence, each 10-mil CRT spot is imaged onto a 20-micron "bit" on the memory plane, which is located about 50 inches from the CRT faceplate.

(3) The memory panel consists of 16 photographic film servo chips and 240 photoconductor-ferroelectric memory chips. The 256 chips are in a 16 by 16 array occupying an area of approximately 1 foot by 1 foot. Two hundred and forty "read-write circuits" can be used to apply 30-volt 2.5-microsecond drive pulses of either polarity simultaneously to the 240 memory chips and to sense the readout signals; alternatively, a smaller number of read circuits might be time shared among the 240 data chips by means of pulse sampling techniques.

With the 15% efficiency of a P22B phosphor, and assuming half the emitted light is lost by reflection and absorption in the optical system, and provided the optical acquisition factor of the system is 3×10^{-5} as shown in the Appendix, the light flux in each channel is then $\geqq 2$ microwatts at an initial beam current of 50 microamperes. To maintain the light flux at the 2-microwatt level, the beam current must be increased to 100 microamperes as the phosphor efficiency decreases due to coulomb aging. This 2 microwatts of light flux per "20-micron bit" equals the 1/2-watt/cm^2 photoconductor irradiance required. At the 50 to 100-microampere beam current and 10-mil spot size proposed, the phosphor laoding would be $\leqq 0.2$ ampere per square centimeter, which is tolerable.[16]

From Pfahnl's work[17] we can deduce that the light emitted from a P22B phosphor decays to half its initial value due to coulomb aging after an accumulated charge of 33 coulombs per square centimeter has been deposited on it. At the maximum beam current of 100 microamperes, this would occur after 165 seconds of accumulated addressing of the same spot. If all 250,000 spots of the system described were addressed with equal frequency, the CRT would last about 45,000 hours at this beam current, considering that the electron beam could be turned on for as little as about 1/3 of each 25-microsecond reference cycle. Unfortunately, in the direct-access usage for which this memory is proposed, some spots would be addressed much more frequently than others. A systems residence application would probably represent the most severe situation.

A rough rule of thumb for systems residence is that 80% of the operations reference 20% of the possible addresses. If 20% of the addresses were to be referenced with equal frequency, the CRT would then last about 10,000 hours at a 100-microampere beam current.

Storage capacity of this 256-channel system would be about 6.0×10^7 bits, the access (cycle) time 25 microseconds (= 5 microseconds dwell time for read-rewrite + 20 microseconds phosphor decay time during which the beam is deflected), and the data rate about 10 megabits per second.

IV. CONCLUSIONS

A memory system technology has been proposed which would utilize existing technologies for all system components except the storage medium. In fact, for the most part, the proposed system would not press the state of the art as stringently as similar previously constructed read-only systems because slower access time and larger spot sizes are proposed for this read-write system. With the slower access time, a highly efficient and long-life phosphor such as the P22B could be used instead of the P16 which was required in the previously constructed read-only systems.

This newly proposed read-write system should also be less expensive than the previous read-only systems because the many costly photomultipliers (and their high-voltage supplies) associated with the read-only memory would be eliminated. It is anticipated that cost per bit would be relatively low in the proposed systems because each component is shared over many bits (i.e., 250,000 bits per lens, ~10^5 bits per read-write circuit, ~10^8 bits per CRT, etc.).

All of the feasibility risk in the proposed technology lies in the storage medium. Developing satisfactory ferroelectric-photoconductor sandwich devices would undoubtedly prove to be a formidable task. However, considering that only a one to two order of magnitude decay in the photocurrent within 20 microseconds is required after the very intense proposed illumination stops, several sensitive photoconductors should have sufficiently fast response time. The use of the photoconductor in conjunction with the ferroelectric should engineer around one of the two most severe shortcomings of ferroelectrics, namely, their lack of a true coercive force. It also appears that the other principal ferroelectric shortcoming of the past, switching fatigue, has been satisfactorily overcome by the development of a new ferroelectric composition described in a previous publication.[2] This new composition also has suitable dielectric constant and relatively small coercive force when in polycrystalline film form. Thus, the task of developing the described memory device should not be unsurmountable.

Principal problems would appear to lie with the interface between the photoconductor and ferroelectric (which might be controlled via a metallic layer too thin to have significant lateral conductivity), with the large current densities which might have to be conducted through the photoconductor (i.e., there may be photoconductor saturation, fatigue, and life problems), and with the dielectric breakdown of the photoconductor.

V. APPENDIX -- Optical System Analysis

According to the Bell Laboratories' analysis,[14] if light is emitted in a "cosine distribution" from a light spot, the fraction of emitted light reaching a spot on the memory plane will be given by the relation

$$\text{flux-at-memory spot} = k_{g_1} k_{g_2} \times \text{flux-from-source-spot}$$

where

k_{g_1} = fraction of incident light transmitted (accounting for the losses due to reflection and absorption) and

k_{g_2} = fraction of light from source spot intercepted by effective aperture of the channel objective lens.

The fraction k_{g_2} is plotted in Figure A1 as a function of reciprocal channel magnification (1/M) and relative optical system length, where the optical system has a length L_S given by

$$L_S = (M + 2 + 1/M)\, f\ell \tag{A2}$$

with

M = channel magnification,

$f\ell$ = lens focal length.

The solid-line curves in Figure A1 have lens f number (= focal-length/effective-lens-aperture) as a parameter. Each set of dashed-line curves has the ratio of CRT raster width to lens focal length, $h_s/f\ell$, as a parameter.

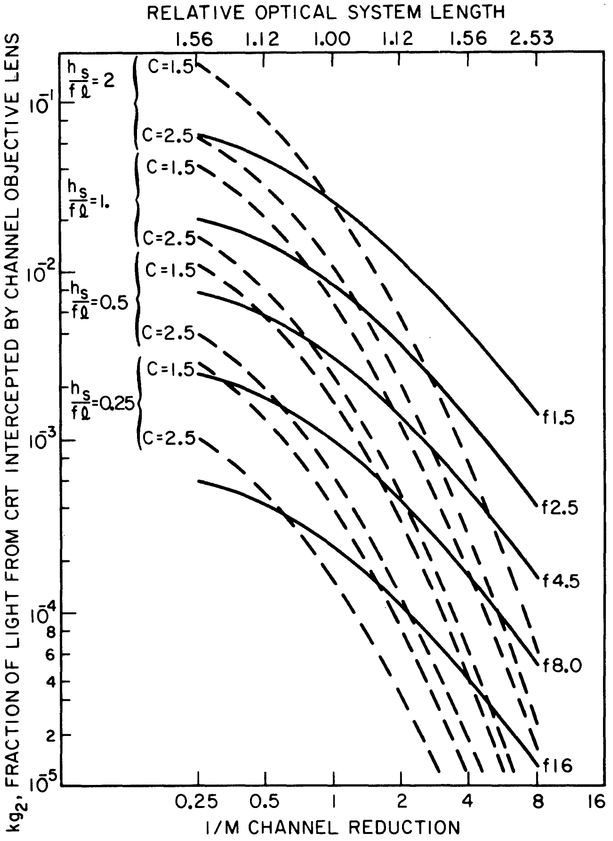


Fig. A1 Minimum value of magnification which can be used for values of the ratio of CRT array size to lens focal length (after Hoover's Fig. 3 in Ref. 14).

The dashed-line curves within each set then have an individual parameter value c which is the ratio of the lens separation (center-to-center) to the effective aperture of the lens. The set of dashed-line curves is inserted on this figure so that a system designer can separate physically realizable systems from systems which would be nonrealizable due to overlap of lenses in the lens plane.

To illustrate, pick a dashed-line curve such as that for c = 1.5 in the set where $h_s/f\ell = 2$, then follow this curve from left to right until it intercepts the solid-line curve having parameter f/8.0. Any portion of the f/8.0 curve to the right of this intersection represents a physically realizable system provided the lens mount, etc., is such that the lenses can be placed as close together as a center-to-center spacing of 1.5 times their effective aperture (c = 1.5).

From the portion of the f/8.0 curve which lies to the right of the intersection with the dashed curve for c = 1.5 and $h_s/f\ell = 2$, we see that if the lens plane is positioned to give an image spot 1/12 the diameter of the CRT spot (i.e., if magnification M is 1/12 giving 1/M = 12), then

$$k_{g_2}$$

will be about 3×10^{-5}.

The one remaining optical system parameter to be specified is the field angle over which the lenses must be corrected for aberrations. This can be determined from Figure A2, where the lens half-field angle which must be obtained in order to get 40, 100, and 250 nonoverlapping channels is plotted on the abscissa and

$$k_{g_2}$$

on the ordinate. Each curve is drawn for specified values of $h_s/f\ell$ and c.

From Figures A1 and A2, it should be possible to achieve

$$k_{g_2} = 3 \times 10^{-5}$$

at M = 1/12 for 256 channels at c = 1.5 and $h_s/f\ell = 2$, with 3.5-inch focal length f/8 lenses corrected over half-field angle $\phi = 15°$. At this

$$k_{g_2},$$

and with a reasonable value of

$$k_{g_1} = 1/2,$$

a 10-mil spot on a CRT faceplate would have to emit 0.14 watt to provide 2 microwatts incident on a 20μ-diameter bit area on the memory plane. Two tenths watt would be achieved by using a 100-microampere 20-kilovolt electron beam together with a P22B phosphor.

VI. ACKNOWLEDGEMENTS

In developing this memory proposal, many valuable suggestions and comments were made by coworkers at IBM, especially J. W. Beck, M. E. Drougard, H. T. Hilton, R. L. Mattson, M. E. Rabedeau, and W. Ruppel. The support and encouragement of O. Kornei is also gratefully acknowledged.

VII. REFERENCES

1. Hoover, C. W., Jr., and Haugk, G., Large Capacity Memory Techniques for Computing Systems (Yovits, Macmillan Co., New York 1962), pp. 79-98.

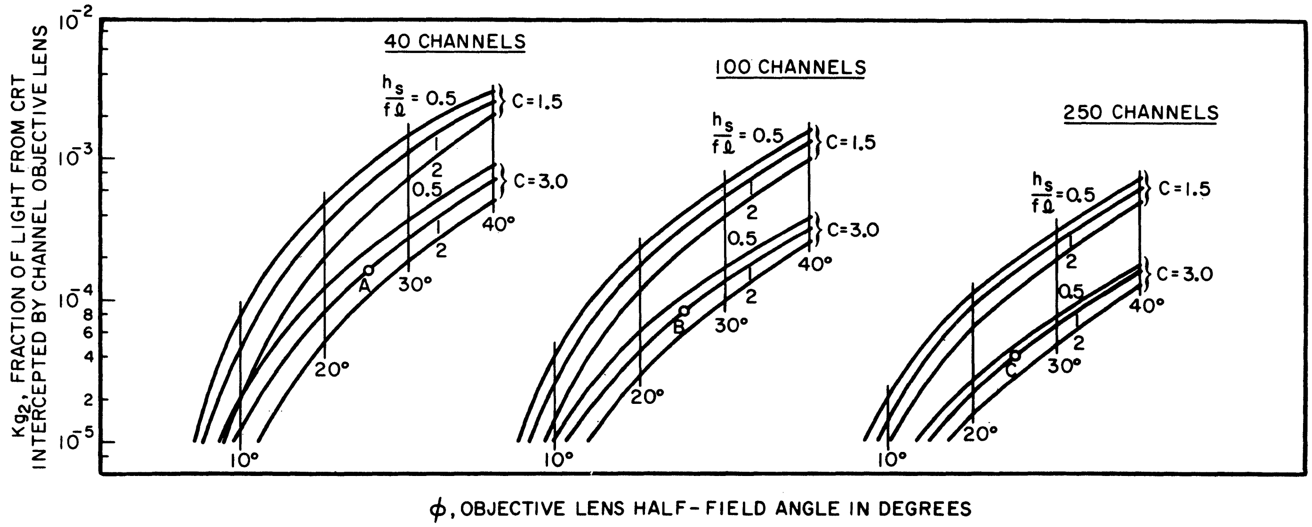


Fig. A2 Field angle needed to get a specified optical acquisition factor in a specified number of channels (after Hoover's Fig. 4 in Ref. 14).

2. Chapman, D. W., "Some Thin Film Properties of a New Ferroelectric Composition," J. Appl. Phys. 40, 2381 (1969).

3. Taylor, G. W., "Electrical Properties of Niobium-doped Ferroelectric-Pb(Zr,Sn,Ti)O_3 Ceramics," J. Appl. Phys. 38, 4697 (1967).

4. Schaffert, R. M., "Photoelectric Recording Apparatus," U.S. Patent 3, 148,354, filed December 20, 1961, issued September 8, 1964, assigned to IBM Corporation.

5. Hanlet, J. M. N., and Requa, S. C., "Ferroelectric-Photoconductive Image Storage," IEEE International Convention Record (USA) 12, Part 1, 38-46 (1964).

6. Sihvonen, Y. T., Boyd, D. R., and Kitts, E. L., Jr., "Analysis and Performance of a Light-Sensitive Capacitor," Proc. IEEE 53, 378-385 (1965).

7. Hell, W. H., and Asam, A. R., "Research Investigations on Photo Facsimile Transmission Techniques," AD 477167, Defense Documentation Center for Scientific and Technical Information, Alexandria, Virginia (January 1, 1966).

8. Heyman, P. M., and Heilmeier, G. H. "A Ferroelectric-Field Effect Device," Proc. IEEE 54, 842 (1966).

9. Anderson, J. R., "A New Type of Ferroelectric Shift Register," IRE Trans. on Electronic Computers EC-5, 184-191 (1956).

10. Sommers, Berry, and Sochard, "Photoelectromagnetic Effect in Insulating CdS," Phys. Rev. 101, 987 (1956).

11. Okimura, H., and Sakai, Y., "Photoelectronic Properties of CdSe Evaporated Films," Jap. J. Appl. Phys. 7, 731 (1968).

12. Merz, W. J., "Domain Formation and Domain Wall Motion on Ferroelectric $BaTiO_3$ Single Crystals," Phys. Rev. 95, 690 (1954).

13. Drougard, M. E., "Detailed Study of Switching Current in Barium Titanate," J. Appl. Phys. 31, 352-355 (1960).

14. Hoover, C. W., Jr., et al, "System Design Considerations of the Flying Spot Store," BSTJ 38, 365 (1959).

15. Cooper, H. G., "High Resolution Electrostatic Cathode Ray Tube," BSTJ 40, 723 (1961).

16. Elliot, W. R., "Limitations on High Energy Cathode Ray Tube Beams with Regard to Phosphor Life," 6th National Symposium, Society for Information Display, New York, September 29-30, 1965.

17. Pfahnl, A., "Aging of Electronic Phosphors in Cathode Ray Tubes," Bell Telephone System Monograph 3989.

SONISCAN - A SONICALLY ACCESSED MAGNETIC-FILM MEMORY

H. Rubinstein, R. Hornreich and J. Teixeira,

Applied Research Laboratory, Sylvania Electronic Systems

ABSTRACT

The Soniscan* magneto-acoustic memory is a new all-electronic mass storage system utilizing a propagating sonic pulse to access a batch-fabricated magnetic-film storage medium. An experimental model of a Soniscan memory system has been constructed. This model had an access time of less than 2 microseconds, a data rate of 7 megabits/line, and a storage density of 4000 bits per square inch. The Soniscan memory is expected to replace electromechanical memories in many applications, since it offers reduced access time, increased data rate, comparable size, and improved reliability, and makes possible new computer architectures. A Soniscan bulk store will be relatively inexpensive, since:

1. Memory planes are batch-fabricated.
2. A single electronic unit, the transducer driver switch, half-selects tens of thousands of bits during reading and writing.
3. Sense and recording electronics are shared among many planes by the series stringing of write-sense lines.

INTRODUCTION

In modern computer systems there is a need for low-cost bulk memories with rapid access times to the block level and high data-transfer rates. The Soniscan magneto-acoustic memory is a batch fabricated, all-electronic memory developed to meet this need.

The Soniscan memory is functionally an all-electronic analog of such electromechanical memories as drums and discs, and can replace them in many applications. A comparison of Soniscan with these memories shows that it has many advantages over them, while still possessing certain of their basic attributes. For example, both Soniscan and electromechanical memories have non-destructive readout and are block organized. Thus, neither requires a large number of semiconductor or other active components to access the memory for recording or retrieving information. On the other hand, Soniscan has a number of advantages over electromechanical memories - faster data-transfer rates, shorter access times, lower power, smaller size, and solid-state reliability.

A typical Soniscan memory system has the following characteristics:

Capacity	60×10^6 bits
Access Time	2 μsec
Transfer rate	14×10^6 bits/sec to 14×10^6 words/sec
Volume	1 - 2 cu ft
Weight	75 lbs
Power	50 watts

Advances in semiconductor technology and computer architecture have significantly increased the speed of modern computers. Memory hierarchies have been developed to take advantage of this speed by combining various memory technologies so as to achieve maximum memory effective speed at a minimum cost per bit. Basically, one or several high speed scratchpad memories (thin film, semiconductor, etc.) with limited capacity are combined with a larger memory, usually core, which finally interacts with some form of mass memory, in many cases a disc. This type of structure proportions the size of each memory in accordance with its speed, cost, and efficiency.

The need for this 3-stage structure arises from the vast difference in access time and data rate between the scratchpad memory and the disc mass memory. The scratchpad can have an access time of 100 nanoseconds and data rates in excess of 300 megabits per second. Discs, on the other hand, have access times of many milliseconds and a data rate in the low megabits per second per track.

The Soniscan memory presented in this paper can eliminate the middleman in the scratchpad-core-disc structure by providing a mass memory that is comparable in price to disc but, being all-electronic, is compatible with the scratchpad memory in access time and data rate.

---

*Soniscan is the trademark for Sylvania's magnetic-acoustic memory

A Soniscan memory consists of batch-coated storage planes and selection and read/write circuits. The memory planes are made by depositing a multilayer thin-film structure on a thin glass substrate and bonding a transducer to one edge. In this structure a copper conductor is sandwiched between two magnetic layers. The magnetic layers are of a magneto-strictive magnetic material; i. e., their magnetic properties are sensitive to externally applied strain.

A Soniscan memory plane has thirty or more data lines, each containing up to 512 binary bits. These lines are simultaneously strained by the scanning strain wave, and each produces a serial data rate of up to 14 megabits per second. For example, if 30 lines on one substrate are addressed simultaneously, information can be recorded or retrieved at a data rate of up to 420 megabits per second. Access time to the first bit on a line is approximately one microsecond, since the memory is entirely electronic and has no rotating parts.

In the following sections of this paper, we discuss the background and theory of a sonic scanning memory, and describe the fabrication of memory planes. We then present details of an operating Soniscan memory model and present a design for a Soniscan bulk store. We conclude with a discussion of the main advantages of a Soniscan memory system over other technologies.

BACKGROUND AND THEORY

Basically, the Soniscan storage element is a magneto-strictive magnetic film possessing uniaxial anisotropy. It is deposited on a glass substrate to which a sonic transducer and write-sense conductor are then attached as shown in Figure 1. Since the film is magneto-strictive, its magnetic properties can be altered by the application of a strain.(1, 2, 3) Such an alteration is illustrated by the hysteresis loops of Figure 2. Comparing the strained and strain-free loops, we note that applying a strain reduces coercive force or threshold for switching and causes a negative or positive flux change, depending on whether the magnetization was in the upper or lower remanent state, respectively.

The former property, utilizing coincidence between an applied magnetic field and a strain, is used to record information in a selected location in the film, while the latter is used to read the recorded information. Since the magnetization will return to its original state after removal of the strain, the reading operation is non-destructive.

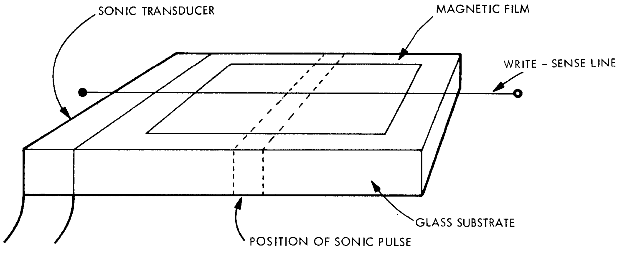


Figure 1. Simplified Soniscan Memory Element

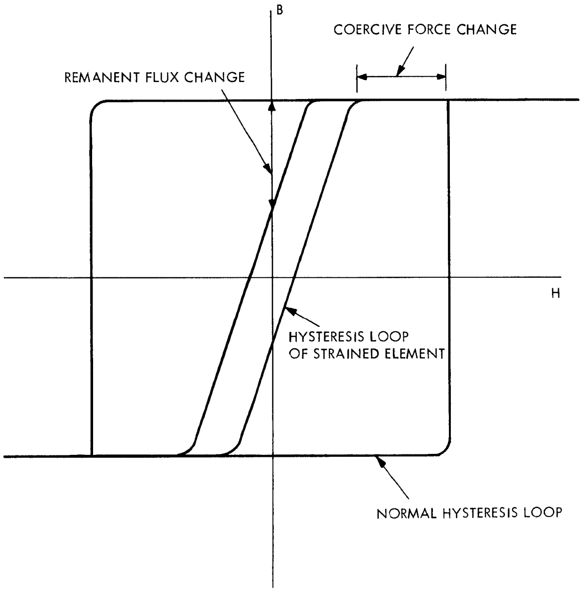


Figure 2. Hysteresis Loops For Soniscan Element

To use these properties, a narrow strain pulse, as shown in Figure 3, is propagated in the glass substrate shown in Figure 1. This strain pulse is generated by applying a voltage to the sonic transducer shown in Figure 1, and has essentially zero amplitude except in a very narrow region. As the pulse propagates, successive regions of the film are selected for reading or writing. Thus, if a series of current pulses is applied to the write-sense winding while the sonic pulse is propagating down the substrate, a sequential recording will be made, i. e., information will be written into the memory. Subsequently, during a read operation, a series of voltage pulses representing the stored information will be generated in the winding by a propagating sonic pulse.

To obtain a high storage density, consistent with reliable operation, it is clear that bit-cell size and interline spacing must be kept to a mini-

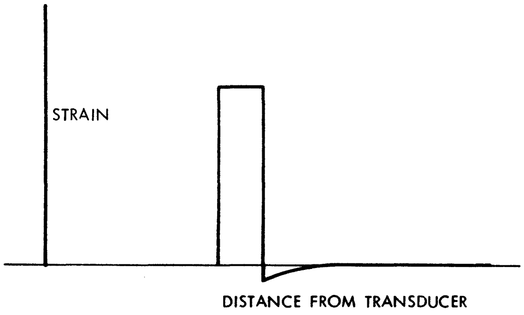


Figure 3. Strain as a Function of Distance at a Given Instant of Time for the Sonic Pulse Used to Scan the Memory Medium

mum. To do this, some means must be provided to prevent the large self-demagnetizing and stray fields that occur in small cells from erasing information stored in the memory. This is accomplished by reducing the magnitude of these fields through the use of a multilayer structure consisting of a pair of magnetic films coupled around the write-sense line. By this technique, bits can be stored densely both along and perpendicular to the direction of sonic propagation. Since the writing current present in any write-sense line will affect only magnetic material in close proximity to it, a large number of write-sense lines can be employed in a memory plane, once demagnetizing effects are reduced. By this means, a high linear density perpendicular to the direction of sonic propagation is obtained.

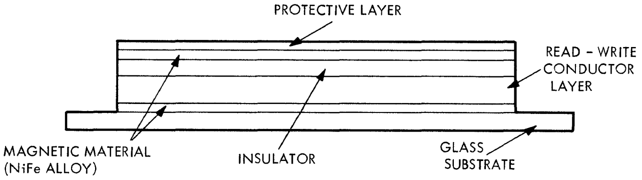
The strain wave employed in the Soniscan memory is propagated as a shear wave in a mode in which the displacement is uniform through the thickness of the glass substrate. The basic reason for using a shear rather than a longitudinal wave is that the chosen shear mode, to a good approximation, is dispersion-free, independent of the thickness of the guiding medium (glass substrate) while no such longitudinal mode is available. A dispersion-free mode is necessary in order to propagate a narrow pulse over any considerable distance.

PLANE FABRICATION AND SPECIFICATIONS

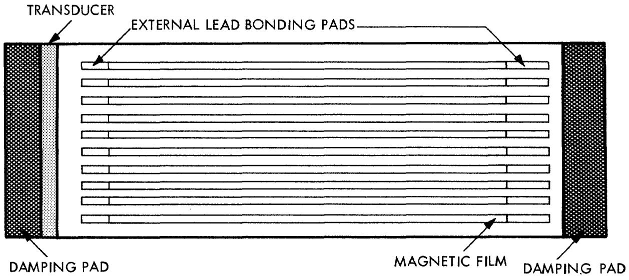
Soniscan memory planes are made by vacuum-depositing magnetic, insulating, and conducting layers onto a glass substrate to form a multilayer film structure, as shown in Figure 4a. The glass substrates used are approximately 10 mils thick and 1 inch wide. The magnetic layers are deposited from a 60%-Ni/40%-Fe melt and are each approximately 1000 Å thick. The conductor layer is copper, and the insulating and final protective layers are silicon oxide. After all depositions are completed, the multilayer structure is etched into the pattern as shown in Figure 4b.

The lead zirconium titanate transducer, in the form of a needle one inch long with a rectangular cross-section, is attached to the substrate edge and the assembly is mounted on a printed-circuit board. Finally, leads are connected to the transducer electrodes and to the write-sense lines, completing the fabrication of the memory plane.

Typical planes are 2 inches long and have 30 to 40 lines, each 10 mils wide and on 20-mil centers. The transducer is 7 mils wide, and requires a voltage pulse of approximately 100 volts to generate the strain pulse. Writing current pulses are about 100 milliamperes with a 70-nanosecond basewidth, and output signals are approximately 0. 5 millivolt, baseline to peak, with a 70-nanosecond basewidth. The data rate is 7 megabits per line, using a return-to-zero recording mode. The output signals are exceptionally noise-free, as no currents are present during the reading operation.



(a) CROSS SECTION THROUGH FILM LAYERS



(b) TOP VIEW FOLLOWING ETCHING, TRANSDUCER AND DAMPING PADS ATTACHED

Figure 4. Soniscan Multilayer Film Structure

SONISCAN MEMORY MODEL

A Soniscan memory model was developed to demonstrate the feasibility of electronically recording, sensing and switching lines on several batch-fabricated Soniscan planes. Line and plane selection as well as read/write functions are provided. Four planes are used in this model, each having 8 (of a possible 30) active lines. Binary-coded logic levels generated by 5 manually operated toggle switches select an active line from one of the 4 planes (32 lines total). Another group of toggle switches set the desired 64-bit pattern to be written on the selected line into a register. The selected line is normally read repeatedly, but a write cycle can be injected by a switch closure when recording is desired. An alternate read/write mode is also available. Total capacity of this model is 2048 bits.

The planes used in the memory model are similar to the one shown in Figure 5. They have 30 lines, each 1-1/2 inches long, with a capacity of 75 bits per line (although only 64 bits are used). The lines are shorted together at the transducer end and are open at the connection end. An extra line on each plane routes the shorted end to the connection end. The planes are mounted on printed-circuit carriers and small jumper wires connect the two together. Lead attachment to the electronics is made from the printed-circuit boards.

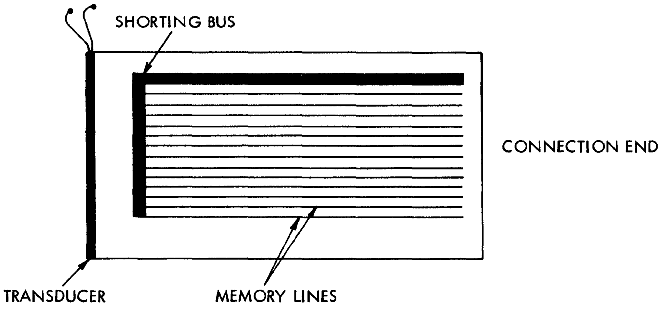


Figure 5. Soniscan Plane Used in the Feasibility Model

Figure 6 is a block diagram of the model. Unique line selection is basically two-dimensional, i. e., the line selection transistor selects one line from each plane, and the plane drivers and the transducer drive switches select one plane from the four. In this model, two line selection diodes are used per line per plane, and they serve for the reset, write, and read functions. Each plane driver uses three transistors for the reset, write, and read functions. The transducer drive switches each contain one transistor. The transducer driver contains a solid-state pulser and the read preamplifier and sense amplifier are two monolithic devices.

Operation of the unit is best illustrated by first defining the write and read modes of operation. In the write mode, the unit enters the bit pattern set in the data switches on the line selected by the setting of the select switches, upon a write command generated by a momentary closure of the write switch. The read mode is the alternative mode in which the unit continuously reads data from the selected line.

A write mode consists of the following:

1. Closure of the write switch causes the events in 2, 3, and 4 below to occur, once for each switch closure.
2. A pulse of reset current generated by the selected plane driver erases the information stored in the selected line without an applied strain. This current flows into the shorted end of the line.
3. The selected plane's transducer is excited by the transducer driver and the closure of the appropriate transducer drive switch. This causes a strain wave to propagate in a direction parallel to the memory lines.
4. When the strain wave reaches the memory lines, near the shorted end, write current from the selected plane driver is injected into the connection end of the selected memory line. The pattern of write current pulses is controlled by the bit pattern set in the data switches. Since the line was initially reset, a pulse of write current is generated only when a ONE appears in the pattern.

A read mode is simply:

1. The selected plane's transducer is excited as above.
2. The plane driver connects the shorted end of the lines on the selected plane to one input of the read differential preamplifier and the selected line to the other input. The data is presented on an oscilloscope display.

A partial schematic diagram of the model is shown in Figure 7. During the reset operation, one of the line selection transistors Q_5 is saturated, placing the cathodes of one diode D_3 from each plane at ground potential. All remaining cathodes of D_3 are at +V. A reset transistor Q_2 on one plane is then saturated, forcing one of the selected D_3 diodes to conduct, and reset current, limited only by R_4, flows through the selected line. Resistors R_3 are large values, and allow only a very small current to flow through unselected lines.

During a read or write operation, one of the line selection transistors Q_6 is saturated, placing the anodes of one diode D_2 from each plane at +V/5 potential. Write current flows through a selected plane by saturating the appropriate transistor Q_3 (note that the write current is in the opposite direction from the reset current). Unselected diodes D_2 are all reverse-biased by voltage divider R_3 during the write operation.

In the read mode, one transistor Q_4 is saturated and a small current is forced through a selected diode D_2 and a diode D_4 by the positive bias from Q_6. Sufficient current flows through the selected diodes to ensure that they appear as a low impedance to the signal path. The preamplifier ignores the DC biasing current at its input, and differentially amplifies the remaining signal components.

The transducer electrically appears as a capacitor, and is excited by the application of a voltage pulse across its two terminals. The pulse from the transducer driver is positive, relative to ground. Referring still to Figure 7, if Q_1 is cut off, its collector and the cathode of D_1 (via R_2) are at +V potential. If V is greater than the amplitude of the drive pulse, D_2 will remain reverse-biased and the transducer will receive no excitation. On the other hand, if Q_1 is saturated, one terminal of the transducer will be grounded and the diode D_1 will pass the transducer excitation pulse.

The bit-packing density on the Soniscan memory model is 50 bits per inch, using a return-to-zero (RZ) operating mode. Typical signal levels are one millivolt peak-to-peak, and signal to total baseline noise is about 10 to 1. (Signal to white noise is better than 50 to 1 - peak-to-peak to rms) The bit rate is 7×10^6 bits per second per line. Figure 8 is a photograph of the memory model preamplifier output (top trace), the logic level output from a threshold circuit, and the write current pulses (center and lower traces).

Performance of the Soniscan memory model clearly demonstrated that the memory elements could be electronically switched at low level with virtually no degradation of signals. The unit was constructed on a large chassis with very little electromagnetic shielding. Figure 9 is a photograph of the unit.

The package houses the electronics, except for the power supplies, under the plate holding the Soniscan memory planes. The electronic circuitry, with the exception of the transducer

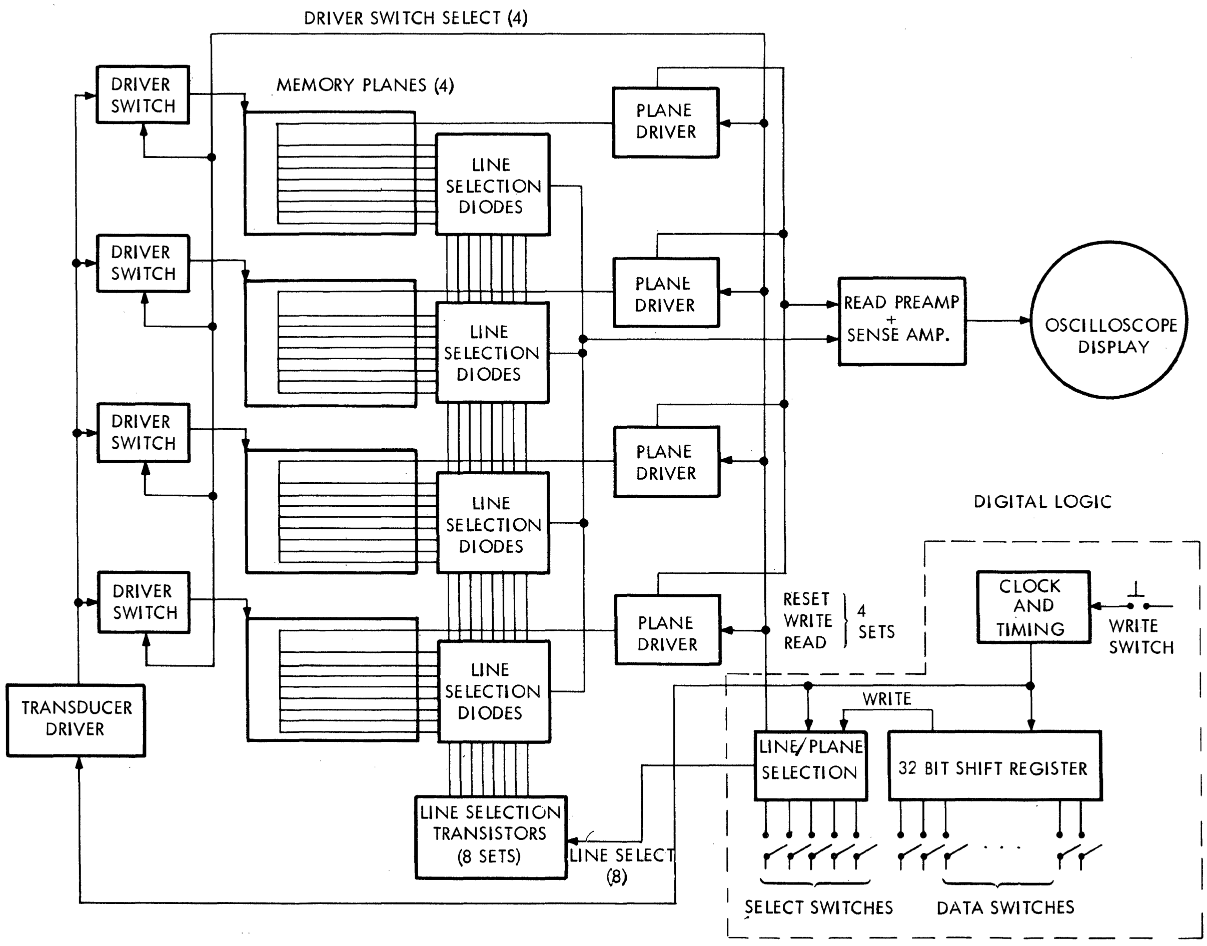

DRIVER SWITCH SELECT (4)
MEMORY PLANES (4)
DRIVER SWITCH
DRIVER SWITCH
DRIVER SWITCH
DRIVER SWITCH
LINE SELECTION DIODES
LINE SELECTION DIODES
LINE SELECTION DIODES
LINE SELECTION DIODES
PLANE DRIVER
PLANE DRIVER
PLANE DRIVER
PLANE DRIVER
READ PREAMP + SENSE AMP.
OSCILLOSCOPE DISPLAY
DIGITAL LOGIC
RESET
WRITE
READ
4 SETS
CLOCK AND TIMING
WRITE SWITCH
WRITE
LINE/PLANE SELECTION
32 BIT SHIFT REGISTER
TRANSDUCER DRIVER
LINE SELECTION TRANSISTORS (8 SETS)
LINE SELECT (8)
SELECT SWITCHES
DATA SWITCHES

Figure 6. Block Diagram of Soniscan Feasibility Model

driver, the transducer drive switches, and the digital logic, is contained on 2 printed-circuit cards totaling less than 30 square inches. The transducer drive switches are mounted separately and require less than 1 square inch each. The transducer driver requires about 8 cubic inches, and the digital logic is mounted on two cards measuring 4 by 6 inches each.

SONISCAN MEMORY SYSTEM

A Soniscan memory will have a low electronic component cost, due to the use of an inexpensive diode switch operating at low level for line selection as in the model. This cost can be further reduced by either increasing the length of the Soniscan planes or by series-stringing several planes to form longer lines. The same pair of diodes then switches more stored information, hence a lower cost per bit. Soniscan planes have been made with an active line length of over five inches, and with both ends of the line available for series-string connections.

Bit-packing density has been tested, using non-return-to-zero recording, at 100 bits per inch on these planes. By stringing 16 planes, each with 512 bits per line, in a configuration as shown in Figure 10, reduced electronic cost can be achieved. Bit-transfer rate in this memory is 14 megabits per second, and data is available in a bit-serial mode. Higher transfer rates in multiples of 14 megabits per second can be realized by duplicating the plane driver and sense amplifier.

The string unit shown in Figure 10, if the planes had 32 lines per plane, would have a capacity of 256,000 bits. Several string units can be used to form a memory module. The string units within this module would share a common transducer driver and line selection transistors.

Data organization in a Soniscan memory is in block form, with a series-string dictating the minimum block length. This restriction is imposed by the fact that a reset pulse applied to a series-string will clear all bits on that

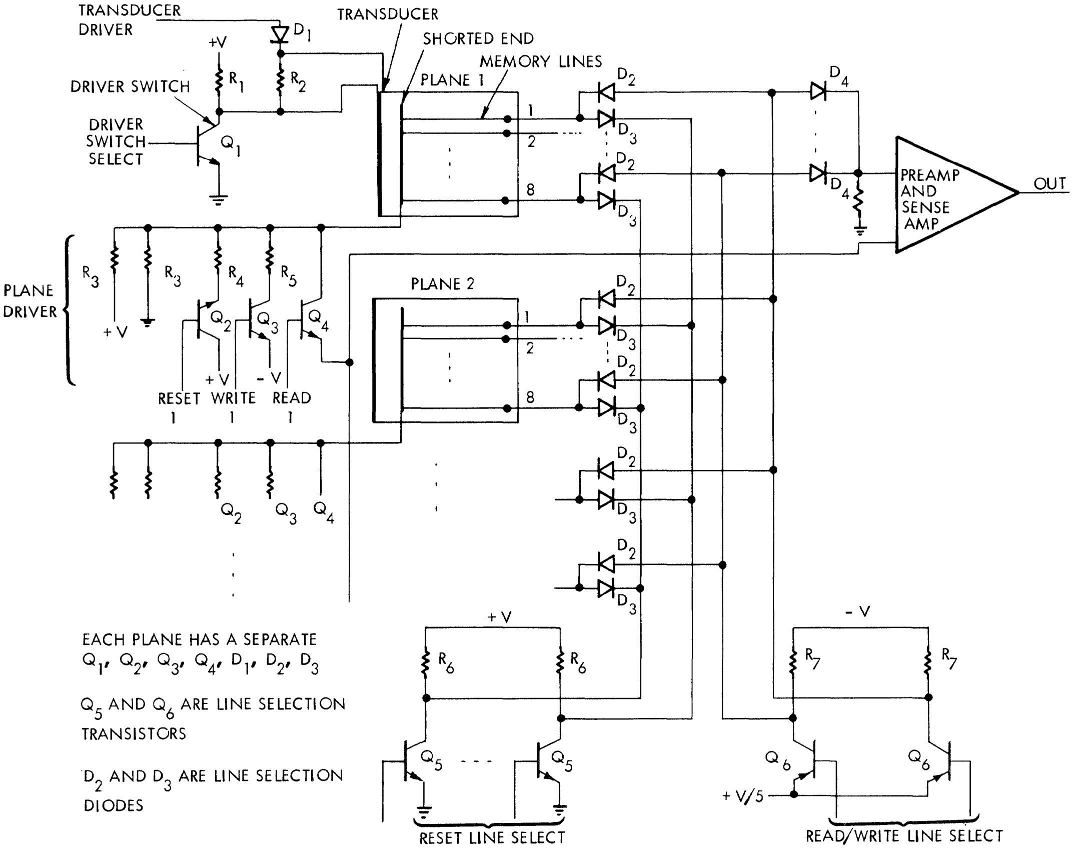

TRANSDUCER DRIVER
+V
D_1
TRANSDUCER
SHORTED END
MEMORY LINES
PLANE 1
DRIVER SWITCH
R_1
R_2
DRIVER SWITCH SELECT
Q_1
1
2
8
D_2
D_3
D_4
PREAMP AND SENSE AMP
OUT
PLANE DRIVER
R_3
R_3
R_4
R_5
+V
Q_2
Q_3
Q_4
+V
-V
RESET WRITE READ
1 1 1
PLANE 2
Q_2 Q_3 Q_4
EACH PLANE HAS A SEPARATE Q_1, Q_2, Q_3, Q_4, D_1, D_2, D_3
Q_5 AND Q_6 ARE LINE SELECTION TRANSISTORS
D_2 AND D_3 ARE LINE SELECTION DIODES
+V
R_6
R_6
Q_5
Q_5
RESET LINE SELECT
-V
R_7
R_7
Q_6
Q_6
+V/5
READ/WRITE LINE SELECT

Figure 7. Partial Schematic Diagram of Soniscan Feasibility Model

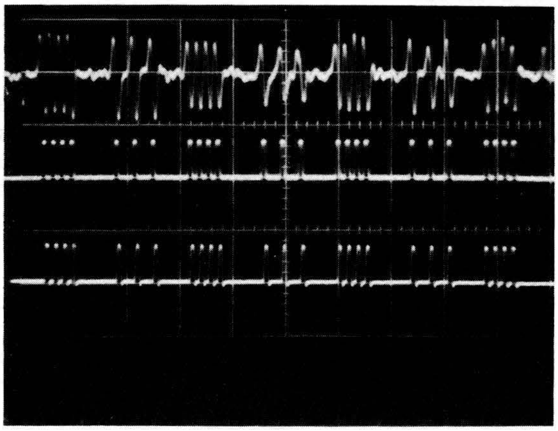


Figure 8. Typical Soniscan Signals (Horizontal Scale - 700 nsec/div.)

(a) Read cycle output (Vertical scale - 620 μv/div.)
(b) Logic level output (Vertical scale - 5v/div.)
(c) Write current pulses (Vertical scale - 200 ma/div.)

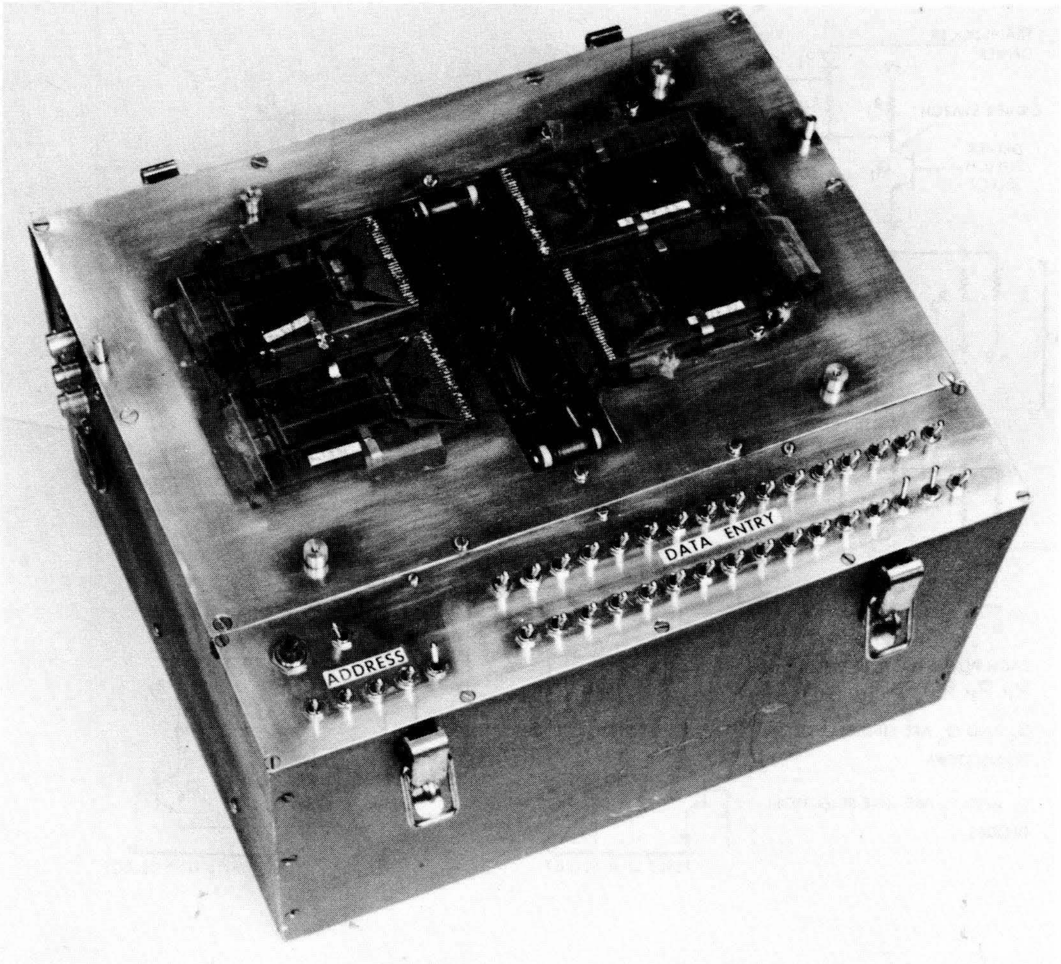


Figure 9. Photograph of Soniscan Feasibility Model

string. Therefore, a write operation following the reset command must write the entire string (8,000 bits). The write function can, however, be done in 512 bit-segments, as can the read function, due to the plane selection (by sonic selection) allowed in the string unit.

Access time to a block is approximately 2 microseconds, the time required to switch a series string and allow the sonic wave to reach the magnetic material, and an entire block of data (8,000 bits) can be written or read in 560 microseconds. This number remains constant, even if bit-parallel operation is used, since all lines on a Soniscan plane are sonically accessed in parallel. Therefore, if all lines on a plane are written or read simultaneously (32-bit parallel) a block of 8,000 words (32-bits per word) can be transferred in 560 microseconds.

ADVANTAGES OF SONISCAN

The advantages of a Soniscan memory system for mass memory applications are numerous. Among the most significant are:

Economy

The low cost of a Soniscan memory is mainly due to three factors: (1) the use of batch-fabrication to make memory planes storing tens of thousands of bits without needing to individually handle each bit, (2) the half-selection of tens of thousands of bits during reading and writing operations by a single electronic unit, the sonic transducer driver, and (3) the sharing without switching of sense and recording electronics among many planes.

Nondestructive Read-out

The Soniscan memory is inherently NDRO; this factor also contributes to lower costs and higher speeds, since rewriting electronics and time are not required.

Low Power

The power requirements for the Soniscan memory are low, as the transducer driver operates at an extremely low duty cycle (less than 1%), writing currents are 100 ma or less, and a small number of electronic components is required.

Minimum Weight and Volume

The Soniscan memory is light in weight and relatively small in volume since it consists predominantly of memory planes produced by evaporating extremely thin films on thin sheets of glass; also, it requires only a small number of electronic components to perform the required switching, driving, and sensing operations.

Modularity

Since a Soniscan memory is assembled by connecting a number of planes (each containing tens of thousands of bits) to their associated electronics, it is inherently modular in nature. Word or block size may be changed at minimal cost, and additional modules may be added at any time without incurring additional costs.

Non-volatility

Since information in a Soniscan memory is

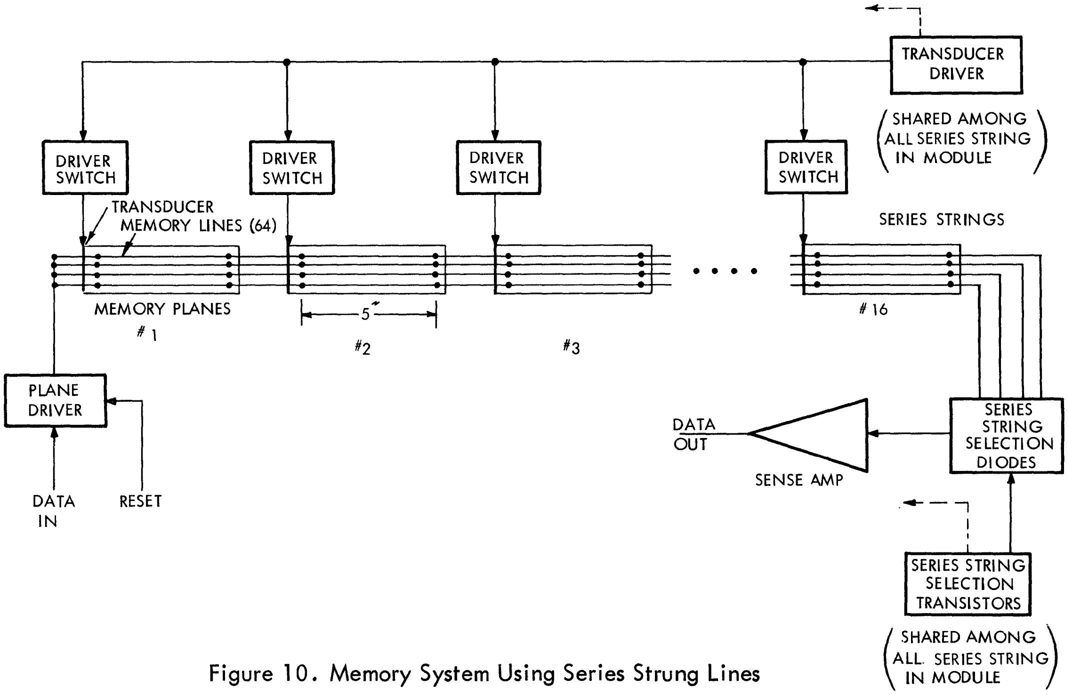

TRANSDUCER DRIVER
SHARED AMONG ALL SERIES STRING IN MODULE
DRIVER SWITCH
DRIVER SWITCH
DRIVER SWITCH
DRIVER SWITCH
TRANSDUCER
MEMORY LINES (64)
SERIES STRINGS
MEMORY PLANES
#1
5
#2
#3
#16
PLANE DRIVER
DATA IN
RESET
DATA OUT
SENSE AMP
SERIES STRING SELECTION DIODES
SERIES STRING SELECTION TRANSISTORS
SHARED AMONG ALL SERIES STRING IN MODULE

Figure 10. Memory System Using Series Strung Lines

stored in a remanent magnetic state, no power is required to store information. Thus, no information loss will occur in the event of a power shutdown.

Reliability

Soniscan provides a non-mechanical analog of drum, disc, and magnetic tape which eliminates most of the reliability difficulties encountered with these block-oriented memories, and thus extends the range of environments in which a block-oriented memory can be operated. Furthermore, the high density along the direction of propagation keeps the number of interconnections small, thereby contributing to lower cost and higher reliability. Again, the small number of required components contributes to reliable memory performance.

Access Time

Since Soniscan is all-electronic, it will have access times to a block of one to two microseconds.

Programming Simplification

By reducing the long delays (access time) needed to obtain data from electromechanical mass memories, simplifications in programming become possible.

Rapid Block Load and Unload

High sequential reading and writing rates can be obtained with this memory. The sequential rate is determined by the speed of the sonic wave and the bit density in the memory. With a density of 100 bits/inch, the sequential rate per line will be 14 million bits per second.

Error Rate

The signal-to-noise ratio on the sense winding is exceptionally good, as reading is done without the application of magnetic fields. To a first approximation, the signal-to-noise ratio is determined only by the amplifier noise and thus is exceedingly high. Low error rates can accordingly be expected.

SUMMARY

This paper describes a new bulk-memory system, Soniscan, that can serve as an all-electronic replacement for drum and disc memories. Soniscan uses a sonic scanning technique to access a high data-density, strain-sensitive magnetic film storage medium. It has rapid access times to the block level and high data-transfer rates. An experimental memory model has been constructed demonstrating the feasibility of the Soniscan concept. It is expected that a Soniscan bulk store will present an attractive alternate to electromechanical units in many applications, making possible computer systems with improved performance and simplified software.

ACKNOWLEDGMENT

We thank Dr. E.U. Cohler and Mr. R.F. Kearns for their assistance in the design of the Soniscan memory system, and Dr. J.E. Storer for his encouragement and inspiration.

REFERENCES

1. J.W. Gratian and R.W. Freytag, "Ultrasonic Approach to Data Storage" Electronics, May 4, 1964, p. 67
2. E.U. Cohler and H. Rubinstein, "Soniscan - A New Memory Device" IEEE Transactions on Magnetics MAG-2, 528 (1966)
3. H. Weinstein, L. Onyshkevych, K. Karstad, and R. Shahbender, "Sonic Film Memory" AFIPS Conf. Proc. Vol. 29 - FJCC 1966, p. 333

THE DYNABIT Ⓣ DEVICE: A VERSATILE NEW MEMORY

B. D. Broadbent., Advanced Technology Department, Hughes Aircraft Company

ABSTRACT

A new magnetic memory device has been designed to fill the gap between core stacks, which have rapid random access to a bit and high reliability, and the less expensive techniques that employ a continuous moving storage medium, such as discs and drums. The DYNABIT^{T} Memory Element is based on controlled domain wall motion through a fine magnetic wire. The device utilizes a modified Ni-Fe Permalloy 0.3 mil wire and operates at speeds as high as 250 x 10^3 bps with the ability to move data forward or backwards at variable speeds and stop instantaneously without loss of the stored data. It maintains all of the attributes of the continuous storage media memories but does not have mechanically moving parts, thus it adds a versatility and reliability heretofore available only in linear select systems. The operation of the memory is pictured as analogous to a static drum surface on which the "bits" move instead of the drum. The simplicity of the basic element structure lends itself to automated low cost fabrication and provides some significant new capabilities that greatly simplify many existing data storage problems. The behavior and theory of magnetic domains is described. General design parameters are discussed, along with some of the tradeoffs available.

INTRODUCTION

The DYNABIT^{T} Memory is a new device based on controlled domain wall motion through a fine magnetic wire. Like other sequential access schemes it precludes the cumbersome and expensive address circuitry of linear select systems. But for the first time it provides an economical alternative that does not depend on mechanical motion. This freedom from inertia provides a versatility that could be matched only by a drum of zero mass and the reliability of one that does not have to move at all.

Present in all memory applications is the cost barrier: the cost of a "standard" configuration must be in the neighborhood of 0.1 to 0.01 cents-per bit to make bulk stores commercially viable. This cost barrier arises because the cost of all-magnetic memories is highly dependent upon the choice of the selection means, which is usually a significant part of the system cost. So, in spite of rapid advances in the technology of plated wire, cores, and thin-film devices, the linear select techniques cannot be considered cost competitive with those that offer less expensive access circuitry.

Among these memory techniques with low circuit requirements are sequential access and mechanical-sequential access memory systems. The most significant for bulk storage applications are tapes, drums and discs. The major cause of failure in present storage devices of this type, and the main environmental limitation in their use, is the requirement for mechanical motion. Furthermore, this requirement to move the entire memory medium past a Read station (which may also be movable) creates a whole new class of access problems, usually summed up in the word "latency". The controller grows in complexity in response to these problems. And whole data processing systems must be designed around the limitations of these peripheral devices.

Typical access times for drums, discs and tapes used in mass storage systems are shown in Table I.

TABLE I

Common Mechanical-Sequential Memory Techniques

Type of Memory	Typical Access Time
Drum	8 - 100 ms
Discs	20 - 500 ms
Tapes	1 ms (closed loop) 1 sec (reel)

It seems clear that using any of the presently available memory concepts to design and construct a 10^9 bit or larger mass computer memory with faster access times would be a difficult task.[1, 2] This is mainly because of the sheer bulk of data and hardware required to insure a high level of reliability. Moreover, the desire for smaller access times and higher computer efficiency is incompatible with the mammoth size that would result if available devices were used in their existing forms. Because of these, and other serious deficiences, Hughes Aircraft initiated research on solid state mass memories. During this continuing research activity, emphasis was placed on magnetic wall propagation within continuous magnetic films and wires. From this work a practical memory, using a fine magnetic wire, was developed. This unique element is called a DYNABIT^{T} Memory.

DYNABIT ^{T} ELEMENT

DYNABIT^{T} is a new memory device based upon magnetic wire wound onto a substrate. Its operation may be pictured as analogous to a static magnetic drum surface on which the "bits" move instead of the drum. The basic simplicity of continuous shifting systems which eliminate the usual cell-to-cell coupling networks and moving parts makes such an approach extremely attractive. For ease of manufacture and thermal design efficiencies, the

---

Ⓣ Hughes Aircraft Company Trademark

substrate (Figure 1) is a hollow tube. Embedded in the periphery of the tube are propagation conductors that run parallel to the tube axis and, hence, at right angles to the direction of the windings. These conductors provide a smooth outer surface to support the wire. The magnetic wire comprises a special alloy whose properties are optimized to provide reliable wall motion characteristics.

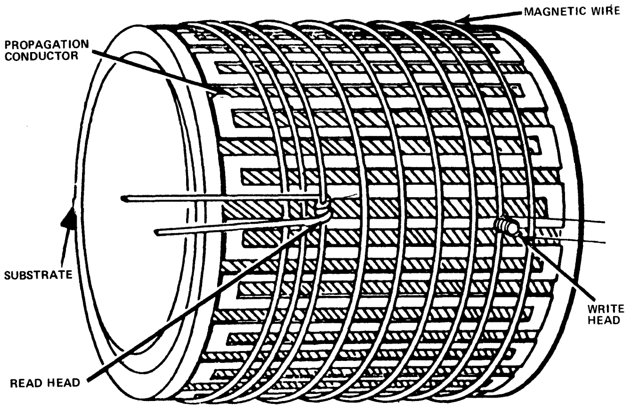


Figure 1 Typical DYNABIT Ⓣ Element

Binary "1"'s are represented by discrete domains. Binary "0"'s are represented by the absence of domains.[3] Detection of digital bits, then, is a matter of sensing for domain wall motion through a Read coil. Signal-to-noise ratios are high because of the squareness of the material. Ratios as high as 150-to-1 have been measured. This ratio is, however, a function of the system configuration.

Several bands of magnetic wire can be wound onto each substrate to form separate storage channels for simultaneous propagation. In one respect, each band of the DYNABIT^{T} Memory represents a shift register of a capacity limited by the length of the wire in the band, but selected by placement of Read and Write heads. The maximum bit rate at present is 250,000 bits per second. But the actual rate may be electronically varied as required by simply changing the propagation pulse rate. The operation may therefore by synchronous or asynchronous. Storage is completely non-volatile, and domains can remain at rest indefinitely until accessed. This also conserves power when the memory is not being accessed. If the system requirement exists, shifting may occur in either a forward or a reverse direction. One may have First-In First-Out (FIFO) operation, First-In Last-Out (FILO) operation, or combinations of these modes.

Readout is effectively NDRO. Normally, after being read, a bit will continue to be propagated until it comes to the end of the wire. This space between the Sense head and the end of the wire permits retrieval of data in the event of power failure. A reverse shift of one step restores the previous data byte prior to loss of power.

There is virtually no maintenance involved in the operation of the DYNABIT^{T} Memory since it is a solid-state device which is relatively impervious to the elements. High reliability is assured because there are no moving parts and temperature coefficients of all materials and parts are closely matched.

The wire and memory elements can be completely tested at every stage of assembly to assure proper operation and increase effective yield (lower cost). The physical simplicity of the DYNABIT^{T} device allows it to be made quite rugged, and it can therefore be used in severe field environments.

Domain Wall Motion:

The DYNABIT ^{T} Memory makes use of the behavior of domains in ferromagnetic material. A number of excellent descriptions have been given concerning this phenomena.[4, 5, 6] The information provided here will form a basis for later description of the DYNABIT ^{T} element. The operation of such a device can be explained, in a very general way, on the basis of a difference that can be made to exist between the energy (magnetic field) required to establish or create a particular domain configuration within a magnetic continuum and the energy required to translate a domain configuration a distance through that continuum. This difference allows "written" domains to be moved unilaterally under the influence of periodically pumped forcing fields without these fields introducing any spurious information.

Consider the classic case shown in Figure 2.

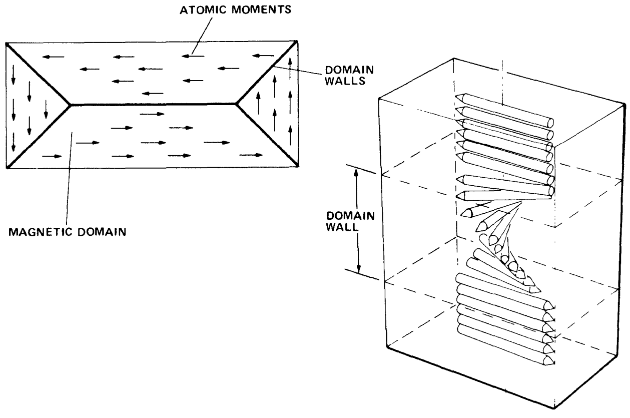


Figure 2. Magnetic Domains — showing Wall Transitions

The arrows represent atomic moments caused by groups of atoms of ferromagnetic material whose exchange forces have caused them to become aligned with each other. These forces are caused by the interaction of spinning electrons in adjacent atoms and act to try to keep the individual atomic moments aligned with each other. However, the localized fields will always try to position themselves such that they reach a minimum energy configuration. The figure shows one such state in which the magnetic moments assume four major orientations to minimize the total energy of the material. Using this figure as a reference, regions containing atomic moments with the same orientation are called "domains". The dark areas, shown by dark black lines, between the domains are referred to as "domain walls". The shapes of the domains will be determined by the internal and external magnetic forces that are present. The total energy of this system must be in equilibrium; therefore, the domain walls must also have a minimum energy or equilibrium state. In Figures 2 and 3 we see a representation of the region of the domain wall. The wall is not an abrupt change of spin moment orientations. Rather, it is a region of finite width over which are distributed many moment directions intermediate to the two domain orientations, and providing a more-or-less smooth transition between the two. From these figures we can see that the atomic moments assume intermediate directions as they progress through the wall. In a very simplified view it can be said that because the atomic moments within the domain wall are already partially displaced, relatively little energy is required to cause the wall to move. A small external field would add to those atomic moments already present to cause those most nearly aligned with it to grow slightly in size at the expense of the others. This small applied field would thus cause domain wall motion.

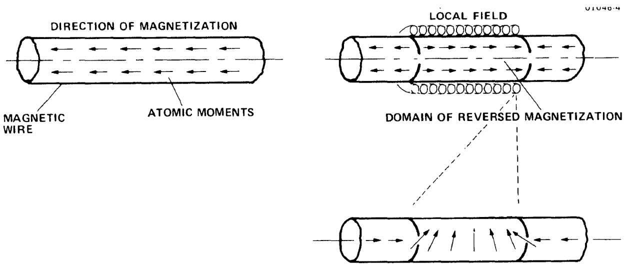


Figure 3. Magnetic Domain and Wall Transition in a Magnetic Wire

We have considered only exchange and external forces in this simplified model. In actuality there are other forces that tend to modify the total field. Two additional significant ones are magnetostatic fields or self-demagnetizing fields as they are sometimes called and magnetocrystalline fields. Magnetocrystalline fields tend to give forces that align the atomic moments into a preferred or "easy" direction. These forces are caused by the crystalline lattice of the particular material being considered. Magnetostatic fields are caused by discontinuities within the magnetic media. These discontinuities can be on the surface or between adjacent crystals. The interaction of all of these forces must be considered in determining the overall characteristics of the wall motion.[7]

The difference between these two energy levels (creation energy and translation energy) may then be inferred from the following considerations: The "creation" energy for establishing a given domain configuration within a magnetic continuum contains conservative terms such as the magnetostatic (self-demagnetizing), magnetocrystalline and exchange energies, as well as the dissipative terms associated with "switching" the magnetic material. The "translation" energy for moving a given domain contains only a dissipative energy term, with the conservative terms remaining constant (provided the domain is translated within a uniform magnetic continuum under conditions that preserve its domain configuration and relation to its environment). It is thus possible in the DYNABIT^{T} element to apply a "propagational" field to the memory medium which is strong enough to move the existing pattern of domains but not strong enough to create new ones. Writing new information is accomplished by a separate Write head which is capable of achieving the higher required field.

DYNABIT ^{T} DEVICE OPERATION

Using the theory of Domain Wall Motion just presented, material has been developed that is highly oriented with its crystal lattice such that it has a preferred (or easy) direction along its length and minimum magnetostatic fields. Such a material will provide stable "domains" oriented with their atomic moments in either direction along the length of the material. Information is stored as a pattern of domains in the ferromagnetic material. These domains are groups of atomic magnetic moments which are all aligned in the same direction, nominally one of the "easy" directions of magnetization determined by the crystal structure. An easy direction is really one representing a local minimum energy condition, although this may not be the lowest energy state possible in the crystal. An atomic moment directed along one of these easy axes will require the addition of external energy to change its alignment in any way. In operation the DYNABIT^{T} Memory elements display a difference between the field energy required to create a reversed magnetic domain and the field energy required to make the reversed magnetic domain expand and/or contract once it has been formed. Information is "written" into the wire by nucleating a reverse 180° magnetic domain. It can thus be seen that there are two distinct magnetic fields applied to the device. One is of sufficient magnitude to nucleate or create (switch) domains in the wire while the other is insufficient to create but sufficient to cause wall motion within the material. These two fields are referred to as the nucleation or switching field (H_n) and the wall motion field (H_w). The area between these two fields represents the band in which the device will move information and hence is characteristic of the operating latitude of the device. If the propagation fields are less than H_w, the walls or information will not be moved, and if the propagation fields are above H_n, spurious walls (or information domains will be nucleated or created. A magnetic material which displays such a differential (H_n-H_w) in its threshold field characteristics tends to switch from one saturated state to the other through the formation of a small reversed nucleus at some point within the media. This nucleus then grows, by propagational switching, (wall motion) to the extremities of the field which initiated the switching action. It can thus be seen that the greater the threshold field differential the greater the operating latitude of the device. One of the critical "figures of merit" for a wall motion device is the ratio of the Write field to the Propagate field. Once the unwanted walls have been excluded from the information channel this Write-to-Propagate ratio becomes substantially equal to H_n/H_w in the simple case. Typical values (for this wire) of these fields are $H_n = 30$ Oe and $H_w = 8$ Oe.

In developing highly reliable memory elements, it is desirable to actually increase the value of H_n. A significant factor influencing the value of H_n is the anisotropy constant K of the ferromagnetic material (2×10^3 ergs/cm^3 typically). In the materials now being employed for memory device operation, the planer intrinsic anisotropy energy E is described to the first order by

$$E = K \sin^2 \theta \tag{1}$$

where θ is the angle between the internal magnetization M and the anisotropy axis (or easy direction) of magnetization. For the class of wires presently being used M is approximately 1432 unit poles per cm^2 (with $B_s \cong 18{,}000$ gauss). K represents the energy-per-unit volume of magnetic material necessary to turn the M vector past the 90° (or hard) direction. Since this is what must be accomplished to nucleate a domain from out of a uniformly magnetized media, a higher value of K is desirable in increasing the value of H_n.

The anisotropy constant K may also be defined in terms of the magnetic field required to turn M into the hard direction. It can be shown that this "intrinsic anisotropy field"[4] H_{KI} is given by

$$H_{KI} = \frac{2K}{M} \tag{2}$$

Along with the development of wire with a large threshold field differential the "intrinsic anisotropy field" has also been increased by appropriate materials processes.

In addition to providing increased operating latitude, a large value of H/H_w is desirable from the standpoint of higher operating speeds also. This is evident from the formula for the velocity v of a wall under the influence of a coercing field H in a material having a wall coercive threshold of H_w. C is a velocity constant characteristic of the material.

$$v = C(H - H_w) \tag{3}$$

This describes fairly well the velocity in the frequency regions in which DYNABIT^T is now operating. It says that the speed of the register is directly proportional to $(H - H_w)$. Thus, for high speed operation, it is desirable to have H_n (switching field) as high as possible in order that the difference $(H - H_w)$ might be made large. It should be noted that the speed of operation on these devices is significantly higher than those cited in previously referenced reports. This is possible because of higher switching thresholds.

When these propagational characteristics of ferromagnetic wire are utilized to form a memory for the storage of digital information, the wire is laid, adjacent to two sets of interlaced propagating electrodes. This is shown schematically in Figure 4. Entering and retrieving information is accomplished through use of the multiple-turn Write and Sense windings shown to the left and right of the diagram respectively. The Write coil is placed such that its axis is orthogonal to the storage wire. This is done as a matter of convenience for the fabrication process. The Write coil can then be placed independently of the winding process and can be placed at any position or positions along the wire. This is not as efficient as if the coil were wrapped completely around the storage wire but the slight increase in write current required is more than compensated for in device flexibility. This configuration provides means for (1) supplying the creation energy to the magnetic wire through application of current to the writing electrode, (2) supplying propagating energy to the magnetic wire through application of current to the propagating electrode, and (3) detecting the flow of domains through the $d\phi/dt$-induced voltage in the readout electrode looping the magnetic wire.

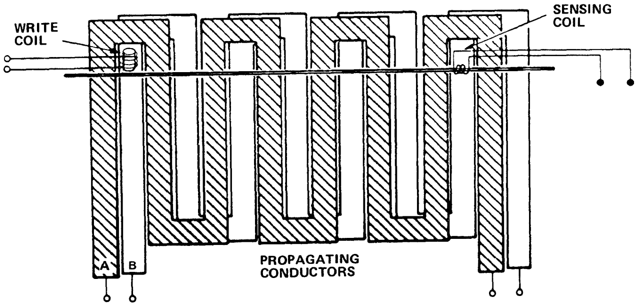


Figure 4. Basic Magnetic Wire Shift Register. Note the multiple-turn write and read winding.

The operation of moving "written" domains under the influence of periodic, continuously pumped forcing fields is shown in Figure 5. This represents a number of side views of the figure just given. Row 1 shows the shifting configuration with its magnetic wire magnetized to the right throughout its length. Current is passed through the writing coil with a direction and magnitude to reverse the magnetization of the magnetic material lying under its influence. This brings about the condition of the magnetic wire shown in Row 2. Row 3 represents the translated state of the domain after current of the proper magnitude and sense has been passed through the propagating electrode set. The arrows indicate the alternating H-field pattern set up by the flow of this current. These fields are seen to be of the proper polarity in the vicinity of the domain walls to induce wall motion in the appropriate direction to translate and preserve the domain. This translation brings the domain into proper position for the next propagating pulse. Row 4 represents the further translated state after the application of current again to the electrode set. For the next translation, a reversed current is required through the

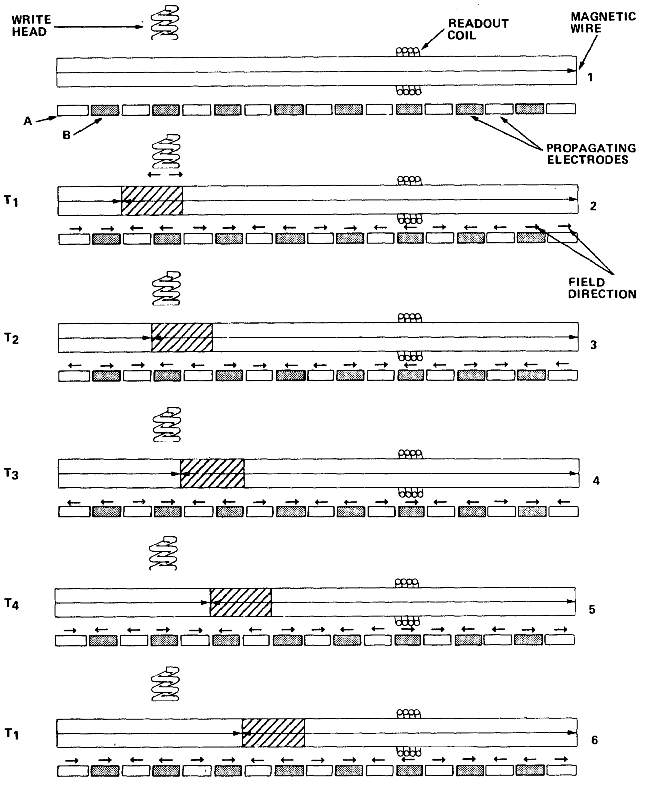


Figure 5. Memory Operation — Cutaway View

ØA Propagating electrode bringing about the configuration of Row 5 and the next translation to Row 6 requires a reversed current in the ØB Propagating electrode. The domain is now in a position for the pattern to be repeated. To analyze the process more closely, we see that during the phase time following the write time of the memory, the propagating fields shift one conductor width ahead. This leaves the leading edge of the stored domain in the center of the field which favors the growth of the domain length in the same direction in which the field shifted. Therefore, the leading edge of the domain again propagates to the new extremity of the positive field pattern. At the same time, the trailing edge of the stored domain is placed in the middle of a field which opposed its existence. Therefore, this edge of the domain is forced to propagate in the direction that will shorten the length of the domain, and thus in the same direction as the leading edge. During each subsequent time period, both edges of the stored domain are propagated simultaneously in the same direction. In this manner, the lengths of the stored domains are retained as they are propagated down the entire length of the wire.

This basic four-beat propagating pattern causes the unilateral flow of domains and may be pumped continuously or stopped at any random point. Because of its sub-critical-field nature, the pattern is effective only in moving established domains and not in creating or establishing new domains. Obviously the direction of information flow may be reversed by reversing the propagating pattern.

Figure 5 shows the memory in cross-section and indicates the field pattern produced by each leg of the propagating electrodes during each quarter cycle time period when the electrodes are driven by a set of square wave, two-phase drive currents. It can be seen from this figure that the field pattern appears to ratchet one conductor width to the right during each quarter-cycle time period. A short interval of time after initiation of continuous propagation currents, the entire length of the magnetic wire will assume the remnant magnetic state in existence at the input end. This occurs because all domain walls within the length of the wire are forced to propagate toward the output end of the memory, thus leaving the wire in the remnant magnetic state existing at the input end. The desired reference state can be established with a variety of biasing techniques.

Information is entered into the memory by driving the Write head to generate the localized field greater than H_n. When properly designed, the established information storage domain will retain its length and shape upon removal of the applied propagating field. The minimum length of magnetic domain that can be used to meet this requirement is limited by the self-dematnetizing field of the stored domains and the magnitude of H_w of the wire. In general, this length must be kept long enough to assure that the static and dynamic stability demands of the individual transition regions at the end of the stored domains can be met without mutual interference. For the class of devices presently being fabricated this minimum length is of the order of 0.040 inches.

DYNABIT ^{T} ELEMENT CONSTRUCTION

The DYNABIT^T Memory Element now being manufactured is a simple low cost technique. For the present medium-density memories under construction, a length of ordinary 5/8-inch-diameter copper tubing is broached with a tool that looks like a splined shaft. This leaves a set of inwardpointing splines inside the tube. These are coated with a relatively hard insulating material. The tube is then swaged with a cylindrical tool to flatten the tops of the splines, forming mushroom shapes. These flattened tops are prevented from touching each other by the layer of insulation. This process is illustrated in Figure 6. Finally, a cylindrical substrate is secured inside the tube, and all the copper except the flattened spline tops is machined away leaving a set of copper strips around the substrate parallel to its axis. To make these strips into two sets of propagating conductors, end caps are put on the substrate that interconnect the strips into two propagating patterns. The magnetic-alloy wire is wound around the completed substrate to provide the final memory element.

The same techniques can be used for the fabrication of high density devices. However, the conductor widths will be reduced to optimize the bit density along the length of the wire. A 0.3 mil wire capable of supporting high storage densities can be used as the magnetic storage wire.

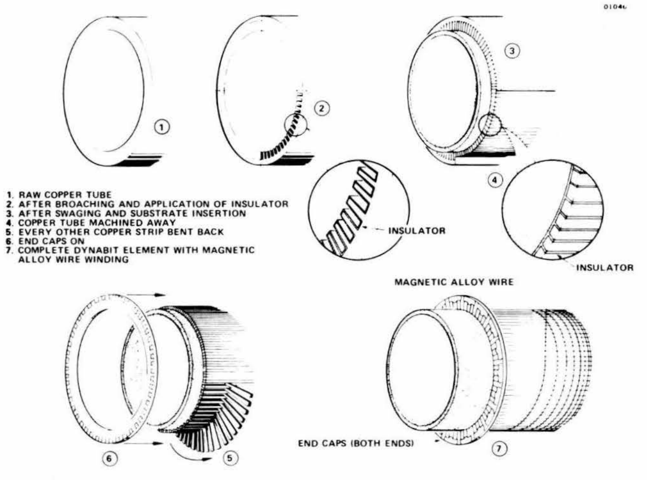

01046
①
②
③
④
1. RAW COPPER TUBE
2. AFTER BROACHING AND APPLICATION OF INSULATOR
3. AFTER SWAGING AND SUBSTRATE INSERTION
4. COPPER TUBE MACHINED AWAY
5. EVERY OTHER COPPER STRIP BENT BACK
6. END CAPS ON
7. COMPLETE DYNABIT ELEMENT WITH MAGNETIC ALLOY WIRE WINDING
INSULATOR
INSULATOR
MAGNETIC ALLOY WIRE
END CAPS (BOTH ENDS)
⑥
⑤
⑦

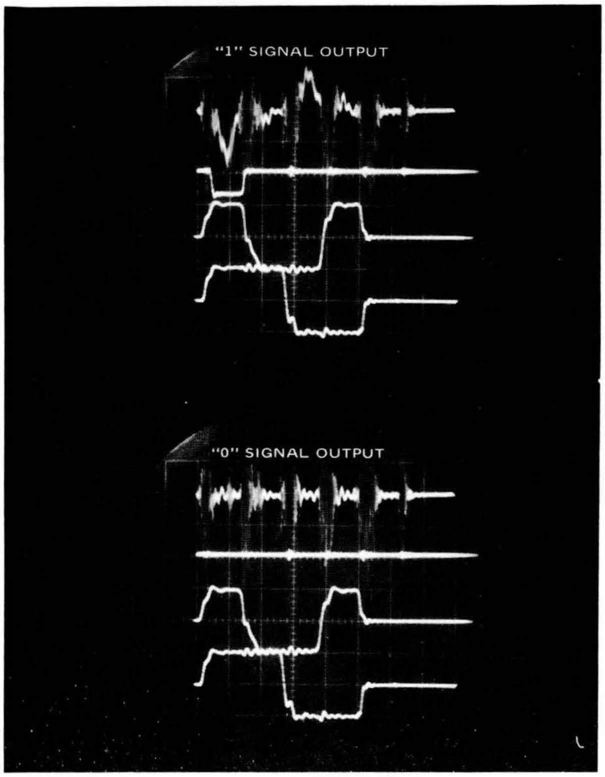
Figure 6. DYNABIT Ⓣ Element Construction

Present wire yield data indicates that the individual storage bands (of the high density memory element) can be 8000 bits long. The larger capacity bands will naturally have a longer maximum access time to the last bit in the band. It is important to recognize that the dimensional tolerances involved are within the scope of normal machine practices. Therefore, the benefits obtained through the small number of manufacturing operations, together with high yields at relatively low unit costs, should be realized as they have been for other memory systems.

ELEMENT CHARACTERISTICS

The DYNABIT^T Memory Element is a relatively simple, completely solid state device. Because of this fact, its potential failure modes are significantly reduced and a high level of reliability may be achieved. Present memories can provide asynchronous operation from a static stored condition to data rates in excess of 200,000 characters-per-second. The current square waves required to propagate the domains are ± 2 amperes and they can vary as much as 20% and still reliably provide the required wall movement. The duration of these propagation pulses is 5 µsec. The Write current required is of the order of 150 ma with a 1.2 µsec duration. The raw output signal is between 4 and 6 mv and has a width of approximately 600 nanoseconds. The rise time is nominally 180 nanoseconds with a fall time of approximately 250 nanoseconds. The shape of the rise and fall times will be varied somewhat by the shape of the propagation current waveforms. The Sense coil itself has a resistance of about 0.6 ohms with an inductance of approximately 82 nanohenries. A photo of typical propagation, write and output signals is shown in Figure 7.

The inherently high signal-to-noise ratio of the basic DYNABIT^T Memory Elements (typically higher than 20-to-1) makes it feasible to monitor several sense windings within a memory array with a common Sense amplifier in most memory organizations. Within an array, the primary source for noise is the electromagnetic coupling between the noise loop formed by the sense winding and the driven propagating conductors of the device. However, since the propagating conductors are laid around the surface of a cylindrical structure, they form a set of interlaced oppositely polarized current tubes. The fields emanating from such a system will be essentially zero at distances greater than 0.5 inch. Thus, the number of sense windings that can be reliably monitored by a single Sense amplifier will be limited by impedance buildup and ground loop problems, rather than excessive noise buildup from the sensing loop itself.



TRACE 1	SIGNAL	5 mV/cm
TRACE 2	WRITE CURRENT	200 mA/cm
TRACE 3	ϕA PROPAGATION CURRENT	2.0 AMP/cm
TRACE 4	ϕB PROPAGATION CURRENT	2.0 AMP/cm
HORIZONTAL TIME = 1 µSEC/cm		

Figure 7. Typical Operating Waveforms

An analysis of the DYNABIT^T element shows that a significant tradeoff exists between power and speed of operation. Changing the data rate from 200,000 characters per second to 2,000 characters, for example, could permit a nominal 100:1 reduction in the average power consumed since very little power is required when the domain is not being moved. The potential failure modes of the memory elements fall into just three categories: those pertaining to magnetization, to environment and to operation (or applied signal). The environmental factor of significance is temperature, since most of the other environmental factors do not affect operation directly but are compensated for in the selection of materials and finishes. Variations in temperature cause expansion and contraction of the components. The maximum temperature range that the DYNABIT^{T} Memory Element can tolerate is limited by two factors: (1) the modulation of the tension experienced by the magnetic storage wire due to an excessive differential between the thermal coefficient of expansion of the mounting substrate and that of the wire, and (2) the modulation of the threshold field and the propagational properties of the magnetic storage wire due to a change in temperature. The first item can be reduced to an insignificant level through the proper selection of the substrate material. Several materials are available which match the thermal coefficient of the magnetic alloy to within ± 1 x 10^{-6} (inches-per-inch-per-degree C). Therefore, a temperature change of approximately ± 140°C is required to produce a 10 percent change in the tension of the magnetic storage wire. The wire composition is such that it exhibits a slight positive magnetostrictive coefficient. Hence increases in tension will increase the ratio of H_n/H_w and decreases in tension will reduce this ratio slightly. These changes are small however over the temperature range listed above. A variation in tension of approximately ± 20 percent due to temperature can be tolerated without serious degradation of the operating characteristics of the memory elements. The temperature evaluations of the magnetic storage wire have indicated that the magnitude of the threshold fields (H_w and H_n) do not change appreciably over a temperature range of -80 to +200 degrees C. Device operating temperature range is now established to normal military environments, e.g., -50° to +125°C, since the real limiting factor is the Curie temperature, which is about 600°C. Other magnetic changes versus temperature do not vary widely over the range of interest.

The maximum shock that a DYNABIT^{T} Memory Element can withstand is limited only by the physical strength of the materials and techniques used to fabricate the elements. Failure from shock would require physical breakage of either the substrate or the Write head mounting bar of the element. Devices have been subjected to a shock loading of 200g's along each of three mutually perpendicular axes. These units were not degraded in any manner by these tests. Though these tests were performed under the non-operating condition, no mechanism is known which could induce errors in the storage of information as a result of shock loading - that is, errors may be induced from the circuits used to drive the memory, but not from changes experienced within the storage elements themselves. Units have also been subjected to a linear acceleration loading of 150g's in all three coordinates for a minimum of 0.33 seconds with no degradation of their operating characteristics. The vibration tolerance is also quite high. This high tolerance stems again from the non-mechanical nature of the device. The typical physical resonant frequency of the memory elements will be greater than 200 cycles. Elements have been vibrated over a 10 to 73 Hz frequency range, with a 0.036 inch excursion, in the non-operating mode, and over a 75 to 2000 Hz range in the operating mode with a constant 3g loading. Each of these tests required that the vibrational displacement occur along each of three mutually perpendicular axes. The demonstrated environmental tolerances of typical DYNABIT^{T} Elements are summarized in Table II.

TABLE II

DEMONSTRATED ENVIRONMENTAL TOLERANCES
OF DYNABIT ^{T} DEVICES

Parameter	Demonstrated Tolerance
Temperature	Minus 80 to plus 200 degrees C
Shock	Up to 200g along 3 mutually perpendicular axes
Vibration	o Non-operating mode: 10 to 73 Hz frequency range with 0.036 inch excursion o Operating mode: 75- to 2000-Hz frequency range with constant 3g loading

The only known catastrophic failures are physical in nature and involve the destruction of magnetic wire or of conductors. These kinds of failures are minimized by suitable protective coatings and by minimal handling during fabrication.

Because data is serially accessed in a DYNABIT^{T} Memory Element, a tradeoff exists between access time to a given word and the capacity of the individual band of magnetic wire. For high storage efficiencies, we would store 8,192 bits in each band. This figure is compatible with several of the physical characteristics of practical DYNABIT^{T} elements, namely, the lineal dimensions which promote good mechanical/thermal packaging designs, yields from the magnetic storage wire, Read-Write head placement requirements, etc. If such an element were operated at a 100,000 bit-per-second shift rate, or at 10 microseconds-per-word, then the average access time to an isolated word would be N/2 x 10 Microseconds, or 40.96 milliseconds. However, if the storage bands were selected to be 2,048 bits in each band, the average access time is reduced to only 10.24 milliseconds. With the band size selected it should be noted that the access time is fixed and independent of the total memory size. The memory size may then be increased or decreased without regard to data access. A typical DYNABIT^{T} organization of this type is shown in Figure 8.

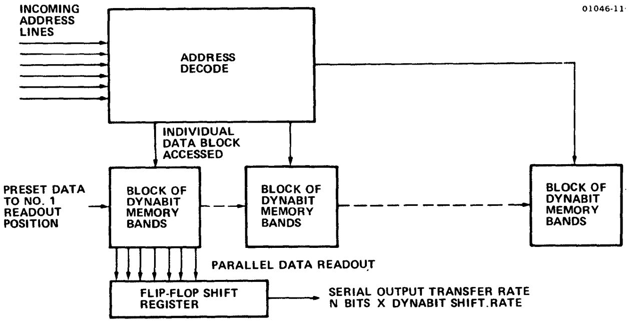

01046-11
INCOMING ADDRESS LINES
ADDRESS DECODE
INDIVIDUAL DATA BLOCK ACCESSED
PRESET DATA TO NO. 1 READOUT POSITION
BLOCK OF DYNABIT MEMORY BANDS
BLOCK OF DYNABIT MEMORY BANDS
BLOCK OF DYNABIT MEMORY BANDS
PARALLEL DATA READOUT
FLIP-FLOP SHIFT REGISTER
SERIAL OUTPUT TRANSFER RATE N BITS X DYNABIT SHIFT RATE

Figure 8. Typical Dynabit Organization. This organization provides for parallel data readout from each block of Dynabit memory bands.

Another variable that can be used in the design of DYNABIT^{T} memories is that of the interrelation of the storage density and the data rate for a given data storage wire. That is, the speed and storage density that can be achieved is strongly influenced by the threshold field parameters and wall motion characteristics of the storage wire.

A self-demagnetizing field is associated with the storage of discrete magnetic domains in the storage wire. The presence of these fields plays a significant role in the operation of the memory elements. The magnitude of the self-demagnetizing field relative to the threshold field of the wire regulates the lineal bit density that can be supported by the storage wire. Also the presence of these fields influences the minimum propagating field demand required to operate the storage elements. These fields are subtractive to the normal propagating fields. Their magnitude increases with the winding pitch of the individual storage helices. Therefore, a high density memory element will require a higher propagating field than a low density memory element. This relationship is shown in Figure 9. In Figure 9A a single turn is wound around a substrate that provides very wide propagation fields because of the wide conductors. In this case the field required to move the domain walls (H_p) would be essentially

$$H_p = H_w \qquad (4)$$

In Figure 9B the domain length in the same wire has been decreased by decreasing the conductor width to a point where the self-demagnetizing field of the domain becomes a significant factor, but the transition still remains stable. Here we see that the field required to move the domain (H_p) is now defined approximately as

$$H_p = H_w + H_D \qquad (5)$$

This is higher than it was in condition A. Consider now Figure 9C. Here a number of turns of wire have been wound close enough to each other such that the self-demagnetizing field of adjacent domains also interact. The field required to move the wall has now increased to the value given below.

$$H_p = H_w + H_D + H_N \qquad (6)$$

In referring to Figure 9 we note that not only has the field requirement for wall movement increased, the total operating latitude has decreased since the switching threshold H_n has not changed. (It is established by the selected wire characteristics.) If high data rates (40 - 250 KHz) are desired, the increase in the operating field demand is established by the velocity characteristics of the storage wire. This increase in the operating field demand must be added to the low speed (less than 40 KHz) propagating field demand (as defined by the magnitude of H_w) and the self-demagnetizing field component (Figure 9B).

The maximum operating condition (storage density and/or data rate) that can be obtained from any given DYNABIT^{T} Memory Element occurs when the propagating field demand approaches the minimum value of H_n found anywhere within the length of the active portion of the storage band (Figure 9C). For reliable operation of a large number of memory elements within an array, the storage density and data transfer rate must be established such that the propagating field demand is kept well below the minimum H_n value known to exist within the length of the wire. Thus it can be seen that a tradeoff exists between storage density (or volume of the memory) and the data transfer rate.

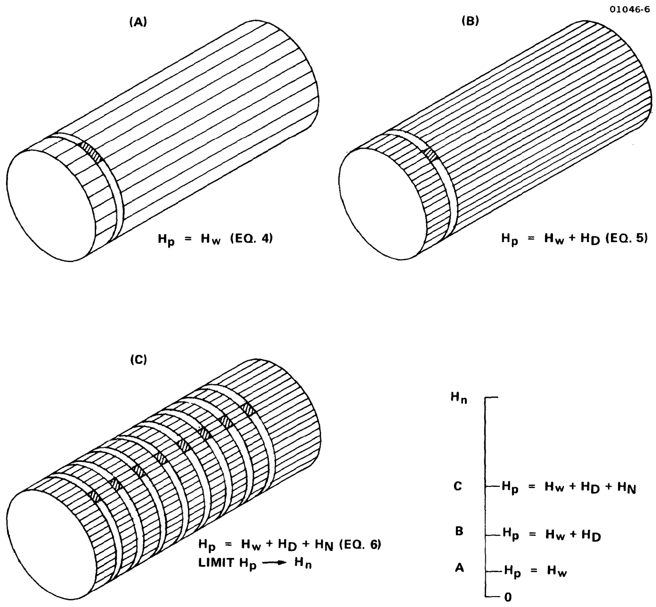


Figure 9. Operating Latitude shown as a Function of Winding Density

The length of wire between H_n minima regulates the maximum storage capacity that can be used in the formation of the individual storage bands. Thus the uniformity in the H_n parameter need not be too great as long as the H_n magnitude remains above a pre-selected value. The magnitude set for the acceptable value of H_n must be a compromise between the system parameters of the DYNABIT^{T} Memory Element and the yield of usable wire. The threshold field parameters and velocity characteristics of wires are capable of providing volumetric storage densities (on an individual element basis) of up to 12,000 bits-per-cubic inch, or data transfer rates of up to 250 KHz. The velocity characteristics of this wire, as described earlier, can be defined by the classical equation: $V = C(H-H_w)$ cm/sec (equation 3). A typical curve for memory wire is shown in Figure 10A. The velocity constant (C) for this wire is typically 13,000 centimeters/sec-per-Oersted of overdrive.

When the velocity field demands are combined with the self-demagnetizing analysis it is found that the tradeoff between speed and volume is significant. This is shown in Figure 10B. As seen from this curve, a 10^9 bit 40 KHz memory will occupy approximately 71 cubic feet, whereas a 250 KHz memory will occupy approximately 118 cubic feet. Thus the six times higher speed device has been bought at the expense of an added 47 cubic

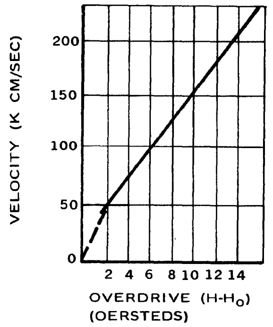


Figure 10A. Typical Velocity versus Overdrive

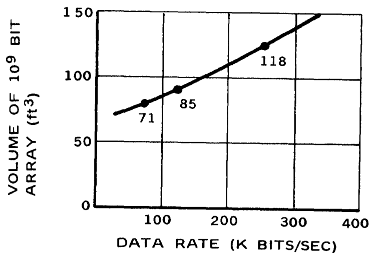


Figure 10B. Speed-Volume Tradeoff Curve for 10^9 Bit Mass Memory

feet of volume. This data is derived on the basis of a 10^9 bit capacity. It should be recognized that these figures reflect volumes required utilizing present packaging processes.

ORGANIZATION

The DYNABIT^T Memory Element is versatile because it can be fabricated with a wide range of characteristics, and elements can in turn be interconnected to produce almost any desired organization. The length and diameter of the substrate tube and the number of turns of wire-per-inch determine the total amount of wire which can be wound on any one element.

The number of discrete bands of wire wound on an element is varied to meet user requirements. Each band has its own input and output; thus four bands means a four bit parallel output, etc. By hooking two, four-band elements together with the propagate conductors in series you have an eight-bit parallel output. Data transfer rates can thus be essentially as high as desired, even though the maximum bit rate of a single band is limited as a function of wire characteristics. The number of bits-per-band multiplied by the number of bands provides the bit capacity of a particular element configuration. The greater the block length (number of bits-per-band) the more economical the memory, as fewer Read and Write heads are required. Control of the DYNABIT^T Memory Elements is simple and flexible, without the complex timing synchronization needed by drum, disc, and tape units. A DYNABIT^T memory controller requires only about 30% as much electronics as a comparable electro-mechanical memory.

The DYNABIT^T Memory lends itself to many applications in the areas of information processing. Because it can function asynchronously, it can work readily into conventional serial logical and arithmetic configurations. Random access to the block level, data pre-ordering for instant access, look ahead without tying up memory control and reverse data flow all enable significant reductions in access time and number of accesses required. The capability to accept and store data at one rate and discharge it at another leads to applications such as buffering data exchange between multiple conductors. Other applications include shift registers, recirculation registers and flip-flop registers, where the DYNABIT^T Memory Element can do the job cheaper, more reliably and with less equipment. Jobs requiring FILO and FIFO can be accomplished without the penalty of reduced capacity present in semiconductor buffers. The memory is truly erasable, making it appropriate for security applications, and DYNABIT^T elements can be removed so that stored data can be transferred from one location to another if desired.

The range of memory sizes to which the DYNABIT^T Memory Element is appropriate ranges from a few thousand bits to billions of bits. Large memories are still easy to mechanize because the element's domains are non-volatile and only a single set of driving circuitry - plus appropriate selection switching - is required.

CONCLUSION

The DYNABIT^T Memory Element can operate in most severe environments with virtually no maintenance. There are no mechanical moving parts, which are the main sources of failures in drum, disc, and tape units. Solid state construction also makes the element inherently resistant to shock and vibration. Almost complete encapsulation and protective coatings make it virtually impervious to the weather. In the area of cost the DYNABIT^T Memory can cover a wide variety of functions with costs that vary from 5 cents-per-bit for special storage and buffering applications to 0.05 cents-per-bit for mass memory applications. A graph showing some typical cost figures for various memory techniques as a function of memory size and access time is shown in Figure 11. This gives a general picture of the area in which the DYNABIT^T memory device will find application.

Between the two points in the spectrum of magnetic memories represented by core stacks (rapid, random access to a bit, high reliability) and the less expensive, compact techniques that employ a continuous moving storage medium (discs, drums, tapes) new memory techniques are emerging. The DYNABIT^T memory device is an extremely versatile, highly reliable, low maintenance, non-mechanical storage device available for this class of memories. Its economic feasibility has been demonstrated. It is presently in production and should join drums, discs, and tapes as inexpensive, compact bulk storage devices of moderate speed and also fill a number of I/O buffering and display applications in a manner that will reduce overall system complexity with improved reliability.

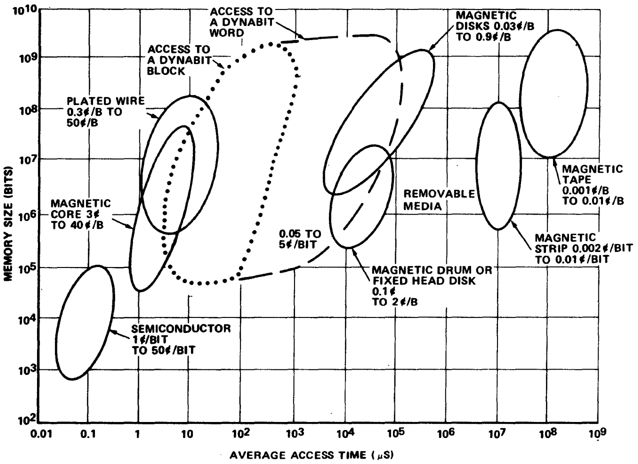


Figure 11. Regions of Access Time and Bit Density of Various Memory Techniques with Approximate Cost per Bit.

ACKNOWLEDGEMENTS

The author wishes to acknowledge contributions to the development of this device by personnel of the Advanced Technology Department of Hughes Aircraft Company, Gound Systems Group, particularly Mr. T. G. Lesher and Mr. H. R. Schultz.

REFERENCES

1. Hobbs, L. C., "Present and Future State-of-the-Art in Computer Memories," IEEE Transactions, August 1966.

2. Freiser, M. J. and Marcus, P. M., "Survey of Some Physical Limitations on Computer Elements," IEEE Transactions on Magnetics, June 1969.

3. Smith, D. H., "A Magnetic Shift Register Employing Controlled Domain Wall Motion," IEEE Transactions on Magnetics, Vol. 1 Mag. 1, pp. 281-284, December 1965.

4. Bozorth, R. M. Ferromagnetism (D. Van Norstand Co., 1961) Chapters 11 and 12.

5. Sixtus, K. J. and Tonks, L., "Propagation of Large Barkhausen Discontinuities," Phys. Rev. Vol. 37, April 1931.

6. Kaenel, R. A. "Functional Domain Wall Propagation Wire Devices for Use in Digital Applications," IEEE Transaction on Magnetics, Vol. Mag. 3, pp. 339-343, September 1965.

7. Bobeck, A. H. and Gianola, U. F., Science and Technology, No. 86, February 1969.

HOLOGRAPHIC MEMORIES

P. Graf, Siemens A.G., Research Laboratories
Munich, W.-Germany

ABSTRACT

Essential components of holographic memories as well as complete memory systems are investigated. Emphasis is placed on write/read procedures based on the properties of light deflectors, storage materials and light receiving elements. Consideration of the single components leads to the design of complete systems. The limits of capacity and access time are estimated and compared with magnetic memories.

INTRODUCTION

In this paper I would like to discuss important properties of holographic memories as capacity, access time and throughput of data as well as their limits due to basic limitations and due to the present state of the art. The development of this new technique started since diffraction limited reconstruction of point sources from holograms has been verified experimentally and since measurements of the signal-to-noise ratio as well as of the diffraction efficiency of holograms[1] assured that high storage densities could be achieved, thereby maintaining sufficient distance to error information. Values in the order of 10^4 bit/mm^2 are obtained today, at least with non erasable recording media, for example with high resolution photographic films. This is about 1-2 orders of magnitude above the storage densities achieved with magnetic materials. Of course, conventional optical techniques as microimaging would lead to the same storage densities, but there are two important advantages, if the holographic principle is utilized: 1) The information is stored highly redundant, because each coherent radiating light source representing a bit is stored as an interference pattern over an extended area of the recording material. Therefore this kind of storing is insensitive against dust and scratches as well as against material defects of the recording material. 2) Imaging is done by the hologram itself, eliminating the need for lenses at least in the read-only-part of the memory. A very important property, common to all optical storage methods is represented by the fact, that coherent light beams, used for reading, can be deflected with very low inertia, thus leading to very short random access times in the order of 1 μsec or even less. This is about 5 orders of magnitude below the access time of magnetic discs. The well known principle of the read only part of the holographic memory[2] is sketched in Fig. 1

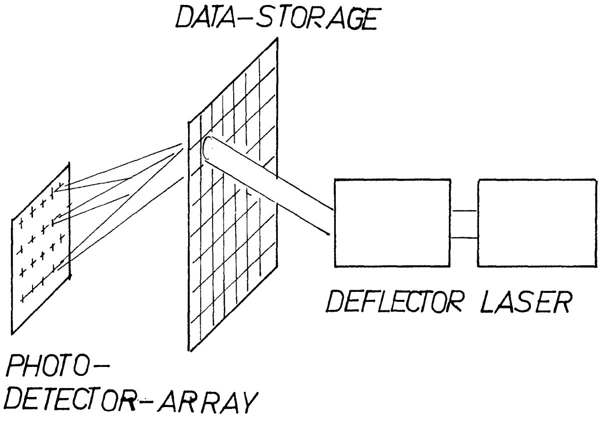


Fig.1: Read only holographic memory

The holographic storage plate is divided into subholograms, which are accessed by a laser beam, leading to the reconstruction of the stored object. All subholograms form their reconstructions at exactly the same place. So, "only" one photodetector array, compatible in geometry with the stored object, is required. Light deflection is done either by electrooptic or acoustooptic means.

Technical use of holographic principles presupposes, that high storage densities can be combined with large overall capacity and that redundance is maintained even in the case of highly periodically arranged objects as the utilized light source arrays might be. The need of a high speed, high sensitive photodetector array with zero redundance is not specific to holographic data stores, but is required whenever information is read in parallel.

Capacity, access times and throughput of data depend on the properties of the system components as laser, light-deflectors, recording materials and photodetector array as well as on their combination to an optimum design. I shall restrict the following considerations to non erasable memory systems.

WRITING PROCEDURES

The basic principle of holographic recording is well known[3]. A coherent wave field containing the information on the object to be recorded is superposed with the so called reference beam, which is coherent with the object wave field. The resulting interference pattern is recorded on a suitable material for example a photographic film. In the case of digital data, the objects are made up by an array of regularly arranged light-sources in on/off-positions, each source representing a binary one or zero. Such objects can be produced either by data masks, coherently illuminated by a laser beam or by means of a light deflector and a single, intensity modulated coherent light source. In the case of masks information is usually stored holographically in a single exposure step in parallel, requiring a different mask for each subhologram. If a deflection system is applied, information is recorded in sequence with a single exposure for each bit, i.e. multiple exposures for each subhologram. Both methods are essentially different from the viewpoint of coherent optics, but thorough analysis shows, that both methods can lead to the same result with respect to hologram efficiency[4]. In practice, multiple exposure serial writing of a subhologram requires enormous stability of the experimental set up. Beyond these principal differences there are physical problems referring to ray geometry, writing speed as well as to the properties of the recording material. They are common to both single and multiple exposure methods and are caused by the requirement of high storage density and high overall capacity.

Ray Geometry

In Fig. 2 a mask M is illuminated by a converging light wave propagating under an angle ψ against the z-axis.

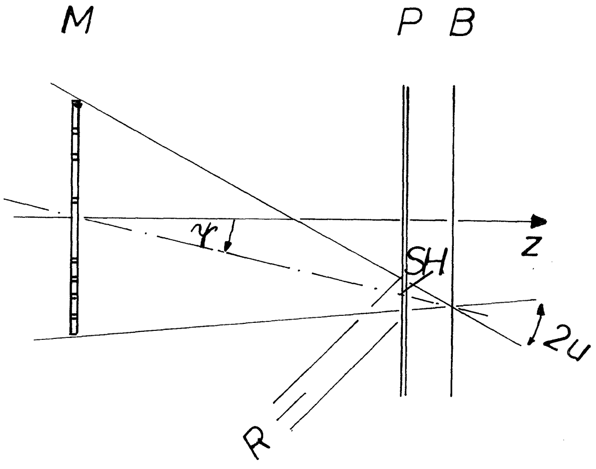


Fig.2: Principal geometry for writing

At SH a subhologram of the mask is recorded on the plate P. The storage capacity is determined by the magnitude of the area of P that can be reached by the illuminating beam and by the azimuth 2u of the data mask (diagonal extension) as seen from the plane B. Because of light diffraction at the subhologram aperture, each stored point source (bit) requires a certain solid angle when it is reconstructed. Thus, storage density m is connected with the azimuth 2u by

$$2\,u = c \cdot \sqrt{m} \qquad (1)$$

where $c = 0{,}61 \cdot \sqrt{2\pi} \cdot \lambda \cdot \delta$
λ means the light wave length and δ is the distance of neighbouring, reconstructed point sources in units of the anular extension of their zeroth's diffraction orders. According to Fig. 2 the overall capacity N is given by

$$N = A\,(u \cdot b)^2 \qquad (2)$$

where $A = \frac{2\gamma^2}{c^2}$

The term γ describes the effective area on P occupied by subholograms. It depends on geometry and arrangement of the subholograms. 2b is the diagonal extension of the storage plate P. The dimension of the whole device has an upper limit, which is given by the maximum size of the photodetector array, if no image forming optics are tolerated in the read-only-part of the memory. It has also a lower limit dictated by the space needed for the reference beam and the requirement for not too different light path lengths between the mask and the different subholograms. It can be seen from equations (1) and (2), that lenses used to form the illuminating beam must have f/D-numbers as low as possible and tolerable aberrations within a large range of ψ. With a given lense, short wave lengths are favorable with respect to high data storage density. Estimations of several designs, based on lense properties, have revealed, that storage capacities of several 10^7 bit can be attained, but that difficulties connected with the lense optics will become prohibitive beyond 10^8 bit. However there are possibilities to replace lense optics by holograms themselves. Instead of forming the mask illuminating beam by a lens, a master hologram can be made, which contains a set of stored illuminating waves. The number of waves, that can be stored follows from consideration of light intensity based on the attainable hologram efficiency*) and on the laser power. Calculations show, that such beam producing holograms should contain not more than about 100 different beams, if short exposure times ($<$ 100 msec) for holographic recording of the masks must be maintained. Production of data storages with more than 100 subholograms requires a set of such auxiliary holograms, which must be interchanged with sufficient speed. Investigation of auxiliary holograms is under way, but the limits of this technique cannot be given at the present moment. Another possibility for high capacity, nonerasable storages is given by synthetic holograms. Computation of the interference patterns connected with special objects as light source rasters and suitable reference beams can be done easily. Because there is freedom in choosing the reference beams the synthetic holograms can be immediately adapted to the properties of light deflectors of the read only part of the system. The decisive point of synthetic holograms is the plotting of the computer results. If the high resolution (several thousand lines/mm) of modern storage materials shall be utilized, other than light optical methods must be applied for physical realization of computed patterns. Limits of such a technique are not recognized at the present moment.

Writing Speed

The writing speed is an additional point but it is not a so very critical one in the case of nonerasable memories. In a lens optic design it depends on the kind of production of the data masks and on the step and repeat procedure, by which the subholograms of the masks are recorded. Existing optical techniques provide for data transfer rates at least in the order of several 10^6 bit/sec. Thus masks up to 100 x 100 bit can be made on photographic film within 10 msec each. Also exposure times for generating the subholograms are within this range, so that the manufacturing speed is essentially determined by mechanical movements of the mask transport. Manufacturing speeds of 1 subhologram/sec seems fast enough in the case of nonerasable memories and can be achieved.

Recording of the Interference Pattern

As is well known, there is only a limited range of linearity in the H-D-curve of a recording material. In recording the interference pattern, the bias point and the degree of the spatial intensity modulation must be chosen in such a way, that a maximum amount of light appears in the reconstructed image and that, ghosts appearing at the sites of the binary zeros in the reconstructed image are sufficiently suppressed. The light distribution at SH in Fig. 2 is determined by the interference of the point sources represented by the holes in the mask and by the interference of this object wave field with the reference beam. As is known from the theory of coherent optics, the light intensity in the object wave field is concentrated in very high and narrow peaks if an overwhelming part of the regularly arranged light sources is in "on-position" and the plate P is located in the plane B. Because the limits of linearity must not be crossed, the very low intensity between the peaks is not recorded. Therefore the effective area of the hologram is reduced to the small part covered by the peaks. Such holograms are of poor diffraction efficiency and very reduced redundance. There are possibilities to avoid this drawback . A very simple way consists in moving the plate P towards the near field of the object waves. An optimum location of P with respect to diffraction efficiency, linear recording and redundance is found experimentally and by computation.

---

*) Hologram efficiency is defined as the ratio between light flux in the reconstructed image and light flux of the reconstructing beam.

READ PROCEDURES

Subhologram Arrangement

In a storage with a given overall capacity N, the information is subdivided by forming subholograms, as has been mentioned. There is only limited freedom in choosing the capacity of a subhologram, if the overall capacity N and the storage density m are fixed. Besides the requirements resulting from compatibility with computer technique there are physical limitations: The properties of the detector matrix, i.e. their size, neighbouring distance and maximum number of the photosensitive elements that can be integrated to a matrix with 100% operation and suitable costs; the amount of available light energy and the tolerable time for reading a single page. The lower limit of subhologram size will be a diameter of about 1 mm, regarding that reasonable distances between hologram plate and detectormatrix amount to 20-30 cm, and on the other hand maximum sizes of integrated photosensitive elements would not surpass a range of 300 μ for reason of electronic signal-to-noise ratio and the necessity of safe operation. If we want to take advantage of a storing density of 10^4bit/mm^2, the minimum number of elements is about 8000. Such devices are not fabricated at the moment with satisfying quality, but they are expected to be realizable in the future with some 10^4 elements, defining upper limits of subhologram diameters between 2-3 mm. The throughput of data is essentially determined by the light intensity at the matrix. A throughput of $2 \cdot 10^{10}$bit/sec seems possible as can be estimated from available power of commercial gas-lasers, the known light losses within the storage system, and the opto-electronic properties of light receiving elements. Thus, reasonable subhologram capacities will amount to about 10^4 bit. They should be accessed within 1 μsec. Storing plates with overall capacities between 10^7 and 10^8 bit will be subdivided in 30 x 30, and 100 x 100 subholograms, respectively.

Light Deflectors

Light deflection systems, adapted to such data storages, must reach Capacity-Speed-Products (CSP, defined as the number of resolvable beam positions times the reciprocal random access time) of the order of 10^8 sec^{-1}. General derivations[5] based on the properties of electrooptic and acoustooptic modulator materials have shown that these values of the CSP can be reached with electrooptic and acoustooptic means. Technical realization depends on whether the optical homogenity of the modulator materials, presupposed in the calculations of the CSP can be really obtained. We have investigated $LiNbO_3$ single crystals as a transverse electrooptic modulator. Due to our investigation, the quality of selected crystals is sufficient for the design of multistage digital deflectors[6] (10 - 15 stages are feasible, i.e. 2^{10} - 2^{15} beam positions. The number of stages depends on the required ratio of signal-to-background light[7]). However, due to the high piezomechanical coupling constants of these crystals, the piezoresonances, occuring usually in the range of several hundred kilocycles to several Megacycles have a very disturbing effect on fast switching. Access times are above 10 μsec at the present moment.

For $Ba_{0,25}Sr_{0,75}Nb_2O_6$ the CSP is 10^{11}sec^{-1}, as calculated from the low half wave voltage of the material[8]. Our investigations of this crystal showed, however, that the dielectric losses are exceedingly high ($\tan \delta$ = 0,3 at 1 Mc) and are therefore prohibitive with respect to fast switching. Besides these difficulties with the electrooptic modulator materials, there are some technological problems connected with the fact that high capacity digital light deflectors (capacity is defined here as the number of beam directions) are always multistage designs. From this viewpoint the acoustooptic method[9] seems to be very simple, because great numbers of well resolved beam directions can be achieved by only two stages. But there are problems too. Deflection occurs by light diffraction at sound waves in the Bragg-regime. The basic principle is illustrated in Fig. 3

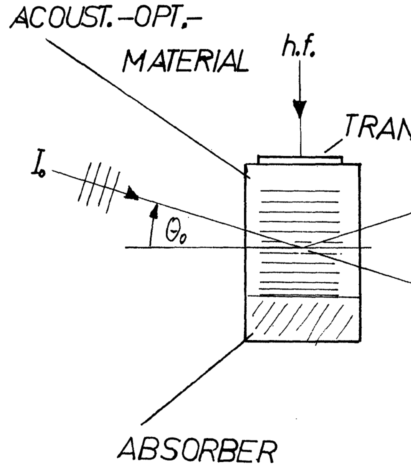


Fig. 3: Light diffraction at sound waves

The deflection angle is twice the Bragg angle θ , given by eq. (3)

$$2\theta \approx \frac{\lambda}{\Lambda} \tag{3}$$

where λ is the light wave length and Λ is the sound wave length, which can be varied by an appropriate electronic and transducer design. The CSP is given by

$$\text{CSP} = \frac{\Delta f}{2} \tag{4}$$

where Δ f is the bandwidth of the whole assembly including electronics, transducer and the acoustooptic medium. The factor 2 in the denominator assures that beam directions are well resolved, their zero'ths diffraction orders touching each other. In the Bragg-regime there is only one diffraction order. Its intensity η , relative to the incident intensity is given by

$$\eta = 5\left(\frac{n^6 \cdot p^2}{\rho \cdot v^3}\right) \cdot \left(\lambda \cdot \cos^2\theta_0\right)^{-1} \cdot \left(\frac{w_0 \cdot P_A}{H}\right) \tag{5}$$

the first bracket of eq. (5) is the figure of merit of the acoustooptic medium. It is determined by the refractive index n, the elastooptic constant p, the sound velocity v and the mass-density ρ . The second bracket contains the light wave length λ and the incident angle θ_0 , the third bracket characterizes the sound field by its beam waist w_0, the field height H and the acoustic power P_A. Eq. (5) holds, if the Bragg-condition is fulfilled. Therefore, if deflection shall take place over a wide angular range, the propagation direction of the acoustic field should be controlled, in order to save acoustic power. There are means to do it. A number of favourable materials with high figure of merit and high optical homogenity are available. Our first investigations in this field were done with water as the acoustic medium. Because of the damping of ultrasound in H_2O at higher frequencies, the bandwidth is limited to 40 Mc. We achieved 25 x 25 well resolved beam postions. A field of 16 x 16 positions is accessed at random with 2 μsec access time by an addressing electronic device. Of course, these results do not meet the requirements of the read-only-memory. So, other acoustooptic materials are under investigation. The solid state acoustooptic deflectors[10] now under investigation will meet the requirements of the sketched holographic memories. -

The comparison between electrooptic and acoustooptic methods reveals that in the range of low numbers of beam directions (< 100 x 100) acoustooptic devices will be more favourable from the view-point of technology, versatility and costs.

CONCLUSIONS

Based on the properties of their single components it was shown, that at the present moment the design of nonerasable memories with an overall capacity of 10^8 bits is feasible. There are several possibilities to extend this capacity limit. It is less obvious that because of fundamental laws in optic it is difficult to reach the storage densities and overall capacities given by the diffraction limit and the properties of the recording material as long as lenses are used within the ray geometry. Our system studies clearly demonstrate these limits and indicate new methods to overcome them. The one element limiting the technical realization of these new memories at the moment is the photodetector array. Comparing the holographic memory with the magnetic memory, one recognizes that both methods have the same range of capacity, but that holographic memories are accessed about 5 orders of magnitude faster than the magnetic memory.

ACKNOWLEDGMENTS

I want to thank Mr. E. Heimbach and Dr. D. Roess for encouraging this paper, and also my colleagues at the laboratory for stimulating discussions.

REFERENCES

1. Kiemle, H. "Phase Holograms in Photographic Emulsions for Digital Data Storage", Optics Technology, May 1969, p.146 - 149

2. Kiemle, H. and Roess, D., Einführung in die Technik der Holographie (Akademische Verlagsgesellschaft, Frankfurt/M 1969)

3. Gabor, D. "Holography - the reconstruction of wave fronts", and Power 12, p. 230 - 234,(1969)

4. Lang, M., Goldmann, G. and Graf, P. "A Contribution to the Comparison of Single Exposure and Multiple Exposure Storage Holograms" to be publishesd.

5. Smits, F.M. and Gallaher, L.E. "Design Considerations for a Semipermanent Optical Memory" Bell Syst. Techn. Journal 46, p. 1267 - 1278 (1967)

6. Kulcke, W., Kosanke, K., Max, E., Habegger, M.A., Harris, T.J. and Fleisher, H. "Digital Light Deflectors" Appl. Optics Joint Issue, p. 1657 - 1667, (Oct. 1966)

7. Tabor, W.I. "A High Capacity Digital Light Deflector Using Wollaston Prisms" Bell Syst. Techn. Journal, p.957 - 990, (1967)

8. Spencer, E.G., Lenzo, P.V. and Ballman,A.A. "Dielectric Materials for Electrooptic, Elastooptic and Ultrasonic Device Applications" Proc. IEEE, 55, p. 2074 - 2109, (1967)

9. Bergmann, L. Der Ultraschall und seine Anwendungen in Wissenschaft und Technik (6. Aufl. (Hirzel) Stuttgart, 1954)

10. Anderson, L.K. "High Capacity Holographic Optical Memory", Intermag Conference, Amsterdam, April 15. - 18., (1969)

VENTURE CAPITAL

Robert F. Johnston
JOHNSTON ASSOCIATES

ABSTRACT

The how, when and where of the availability of high-risk money for new computer companies are presented in this position paper for a panel discussion. The objectives are to define factors that an investor should consider and factors that a company should consider. Panelists included Emmit Noland, Vice President of Loeb, Rhoades and Co.; Daniel Sinnot, President of Interdata; and Tim Collins, President of Collins Securities Corp.

Venture Capital is high risk money with a principal interest in capital gain which has traditionally come from wealthy individuals, pools of family capital from such families as the Rockefellers, the Whitneys, and the Phippses, and from a few public funds such as American Research & Development (ARD). The last two to three years have seen an increasing interest on the part of the more aggressive money managers at both banks and insurance companies to put a portion of their assets into this area and to take a long-term outlook on their commitment to the venture capital field.

A secondary source of money for the venture capital field is the individual investor with a net worth range of between one million and five million dollars. These people are normally investors in high risk situations with money which they have just made in the stock market. For example, an individual in this category might have owned a stock such as University Computing and watched it go from $20.00 to $160.00, taken part of his profits from that investment and reinvested those profits in a small software consulting firm, with the thought that he could find another University Computing and be one of the original founders. The availability of money from this source is very much contingent upon the performance of the overall stock market, since normally such an individual investor has limited experience in the venture capital field with starting small young companies. For example, when University Computing stock goes back from $160.00 to $60.00, the investor is hesitant to put additional money into high risk start up situations.

Thus, the availability of venture capital money has a greatly accentuated curve in relationship to the swings in the stock market. With this availability, the corresponding terms of this money are greatly accentuated. The willingness of the public to buy unseasoned companies in the "new issue market" also tends to accentuate these fluctuations and to beguile many investors and management groups into a false assumption regarding the ease of raising money.

CURRENT CONDITIONS

The market conditions of the last year have witnessed a dramatic portrayal of the aforementioned cycle, particularly in the computer industry. The investing community decided approximately two years ago that companies in the computer field could only grow and be profitable. Hence, they bid the stocks of the public companies to ridiculous heights and found money for the creation of many companies which should never have been in business. Within the last six months, the combination of tight money, a falling stock market, failure for many of the companies to meet their projections, and a reversal of the investors' attitude will likely precipitate a series of bankruptcies among small companies in the computer field. The situation is very analogous to the same type of fluctuation the electronics industry went through in 1961-1962.

CRITERIA FOR OBTAINING MONEY

Even under the adverse market conditions currently existing, there is money available in the venture capital field for the good management team with a soundly conceived idea. The following are the major criteria which a venture capital investor wishes to evaluate. The first and most important criterion is the quality and track record of the management team. What have they done before? Do they have within their group the complimentary skills of management, technical know-how, and marketing?

Secondly, how unique is the idea? To what extent can it be protected from someone else duplicating what the company is doing?

Thirdly, how well has the company thought out the problems of accomplishing its venture?

The investor prefers to obtain a proposal from the management team which includes an explanation of the idea, financial projections, and detailed resumes on the principals. In this instance the investor is principally interested in the reasons for the numbers in the financial projections.

SOURCES OF ADVICE

Assume you are the president of a small company. You either have a new idea or possibly your company has been in existence for a while and wishes to raise additional capital. How should you go about raising such additional capital? There are several traditional sources of advice regarding financial affairs and the raising of capital. Your accountants are frequently in touch with people who either have money or who know of sources of money. In addition, your banker and your lawyer should be asked for advice in this area.

The following are the criteria the management of the company should weigh in deciding from whom they will obtain the capital:

1. The source of money should be professional investors in the venture capital field. In other words, they should already have done several high risk investments over the past three to four years. Eight out of ten venture capital investments either go bankrupt or get merged out under unfavorable terms and it is important that your investors be familiar with the problems of this area and the risks involved. Unfortunately, many of the wealthy individuals who are playing with extra money in the stock market assume that if University Computing could start with one thousand dollars, you certainly should be successful with one hundred thousand dollars.

The second reason for dealing with professional venture capital investors is that they are aware that they will probably have to put up additional money and are prepared to do so. This is not to say professional venture capital investors always put up more money, but they are normally in a better position to evaluate the mistakes and to be constructive with suggestions as well as money. Because these people are usually exposed to a much wider range of investment alternatives, they are more selective and frequently require more advantageous terms than the less sophisticated investors. The management team has to weigh this differential of terms against the more professional approach of the venture capital investor to management problems, along with the additional ease of raising money at a later date once there is a recognized professional investor in your company.

2. Important also is how comfortable you feel with your investors. The relationship between the investor in a small privately held company and the management team is somewhat analogous to being married. Both parties are forced to go through many trials and tribulations together but there is enough room for disagreement in the explanation of missed projections, incomplete financials, incomplete information regarding management, without the additional complication of mistrust on either party's side. Because eight out of ten venture capital situations have difficulties, the management team is in the position of explaining why they didn't make their projections. There is an obvious need on the investor's side for confidence, honesty, and forthrightness of the management. There is an equal need for trust in the opposite direction because as the company acquires additional capital, management should not feel that it is being taken advantage of. Obviously, if the company has missed projections and run out of money, it is in a very poor negotiating position in terms of additional capital.

EVALUATION OF TERMS

In searching for money, the management team has to consider not only from whom it will obtain its capital, but under what terms; i.e., what percentage of the company will they give up for the money. The answers to these questions are generally a function of the overall psychology of the market and particularly a function of the specific investing group toward your project. Along with the question of what percentage of the company is the corollary problem of the financing structure of the company. Do you issue warrants to purchase the stock at a slightly higher price in the future, is the stock held in escrow under a performance contingency for the management, or is there a right of first refusal for future financings for the investing group. The structure of these options can have serious implications for the next stage of financing, and therefore should be carefully thought out by the management of the company and their advisors.

When we previously discussed sources of advice and mentioned accountants, lawyers, and bankers, we failed to include a third group who are the people whose profession it is to advise young companies specifically on these problems. These people or groups of people frequently are retained by the company to (a) advise it on the financing structure, and (b) locate the money for the company. These people have the advantage of knowing the implications of various alternative financing structures and can advise the company on such implications, different alternatives in terms of future financings, as well as considering alternatives which might be more saleable under the current market conditions.

These people are alternatively called promoters, finders, consultants, and several other non-printable terms. It is normally to the company's advantage to pay them in a combination of cash and stock warrants to insure their continued interest and advice. The best way to evaluate the professional money raiser is analogous to one's criteria for the investing group. Management should be comfortable with the person on a personal basis and should be familiar with one or two of his previous financings and his current relationships with those companies.

SUMMARY

The current market conditions are difficult for obtaining venture capital in the computer field, but money is available and the potential gains on successful projects are still very obvious to the members of the computing industry.

MAIN FRAME MEMORY TECHNOLOGY
A DEBATE

T. R. Finch, Chairman
Bell Telephone Laboratories, Incorporated
Murray Hill, New Jersey

Core memory designers are quiet with passive confidence that their dominance of the memory market will continue into the next generation main frame systems. The speed/cost ratio improvement of several hundred to one that has accrued during the 60's to system designers using ferrite core systems makes the core a rapidly moving target for any new technology. The core memory designer intends to keep it this way and history of proven performance is on his side.

The plated wire technologists state that 1970 heralds the beginning of erosion of core dominance. Higher speed, NDRO capability, lower power provides a flexibility for factoring a variety of system storage needs with plated wire that core cannot match. Plated wire lines are now producing the required material uniformity to meet the production volumes required for competitive costs. They are ready!

Also, 1970 seems to be the year of fruition for silicon LSI which has been repetitively promised by very active, vocal semiconductor technologists. And memory is the competitive vehicle. Improved processing, outstanding photolithography, refined circuit configurations have resulted in active memory cells providing array density outstripping both core and plated wire. Exceptional speed, NDRO, logic compatibility, and low cost module modularity are the stepping stones for the adaptability to provide universal memory potential that magnetics cannot meet. So say the MOS and bipolar semiconductor memory technologists. Thus the stage is set for the great memory debate.

Will the next generation main frame memory be magnetics or semiconductor. Leo Slattery of Fabri-Tek believes it will continue to be magnetics. Dave Hodges of BTL says definitely not, the attributes of semiconductors provide system trade-off potential that will dominate in the 70's. They will present their positions and rebut each other's arguments for the first debate.

If the next generation system is magnetics, will it be core or plated wire. Phil Harding of Electronic Memories says the core will continue its evolutionary power, but George Fedde of Univac believes not, the potential is with plated wire. Thus the second debate.

If the next generation system is semiconductor, will it be a single device technology, either MOS or bipolar depending upon application, or a hybrid approach combining MOS and bipolar. Gordon Moore of Intel believes single technology will win out in most applications. But Wally Raisanen of Motorola believes the trade-off performance-cost power will favor the hybrid design combining MOS and bipolar. Thus the third debate.

Such activity and competition will have considerable impact on system design, and memory hierarchies will be affected. Thus emphasis will be placed on the properties of interest to system designers, rather than on the material-device structure considerations, in order to forecast future acceptance and probable use.

The readers of the following papers and the audience listening to the debates should realize that in almost all cases, the authors represent organizations which have interest and activities in all technologies so they are not asked to represent company or system commitments. Rather, they will make every effort to delineate the specific properties of a particular technology and assess their worth in memory.

MAGNETICS VS. SEMI-CONDUCTORS AS THE MEMORY ELEMENTS OF THE FUTURE

Leo F. Slattery
Fabri-Tek, Incorporated
Memory Products Division
Minneapolis, Minnesota

INTRODUCTION

Through all of the 1960's magnetic storage elements have been the predominant memory devices in all classes of digital computers. The most promising non-magnetic contender for random access computer memory is the semi-conductor memory which now appears to be on the verge of technical, if not commercial feasibility. An application niche seems assured for semi-conductors memory elements in small-sized and high speed memories. Such memories find use in terminals, paging, control processors and cache duty. The depth of semi-conductor penetration into core territory will depend largely upon price. Core memories are an elusive cost target due to continued improvements, foremost of which are trends toward 2 and 3 wire stack designs, radical unitized packaging and the use of micro-electronics. Most current semi-conductor memory prices are in the form of projections made by officers of recently formed companies operating on venture capital or by R & D people actively engaged in exploring application possibilities of semi-conductor memories. The large majority of the non-moving random access memory cells produced during the next 5 to 10 years seem destined to remain cores. The strongest present contender for partnership with the core as a practical magnetic storage element is the plated wire.

MAIN FRAME MEMORY APPLICATIONS

There can be no doubt that the most attractive application area for memory devices of any sort is that of random access memory. By this is meant memory accessed with a minimum latency delay by means of a single address for the purpose of reading out of or writing into any address. The essential elegance of such a memory has led to its dominance in the central architecture of the modern digital computer. The cost of memory of this type has been the most important limitation on its application.

The insatiable demand for memory and the finite availability of money have led to the development of second-order memory with a large increase in access time. Such second-order memory, exemplified by tapes, disks and drums utilizes moving or rotating magnetic surfaces. Only the most visionary could possibly see these second-order memory uses a potential application area for micro-electronic memory.

During the 1960's the dominant device for main frame, random access memory has been the ferrite core. Early in the decade this device supplanted the cathode ray storage tube, the drum and sonic delay lines as the preferred storage element. Except for momentary scares provided by cryogenics and thin films, cores have sailed serenely through the whole decade.

The past year has seen a vigorous struggle for computer memory application made by semi-conductor protagonists. It is possible to spell out some of the functional requirements and cost goals that semi-conductor (or any memory element) will have to meet in order to obtain wide application. While it is true that if a device is cheap enough many other undesirable characteristics will be tolerated, the application niche of random access, read-write main memory is classically defined and represents the "main-chance" for semi-conductor memory or any other contending memory device.

The target is an attractive one because of its dollar volume and its relative hospitality toward Large Scale Integration. In contrast to the relative complexity and non-repetative uniqueness of computer control circuitry, the monotony of word after word of memory circuits proves seductive to large-scale diffusion advocates. The simplicity of signal interconnection between memory and the rest of the computer is unequalled in any other application. Twenty address leads and 40 data leads plus a few power and status leads allows one to communicate with millions and millions of storage elements. The dollar volume of current main memory useage is apt to stagger anyone on the outside looking in. However, in looking at this billion dollar potential market it should be borne in mind that the large computer manufacturers themselves dominate the design, application and manufacture of main frame memory. They have the money to duplicate any technology, hire development crews and master any process. In this day and age large computer companies will not allow themselves to be exploited by mere inventors. In sum, then, the issue is fairly drawn; the most active LSI protagonists have selected their most attractive target and they are doing their thing.

MINOR APPLICATIONS

Certain applications of semi-conductor memory devices, while they may prove profitable, do not constitute a threat to magnetic storage device (core) dominance. Once again, this is because the "main-chance" is random-access main memory.

To be specific: solid state shift registers, MOS or otherwise, are barking up the wrong tree and will be consigned to desk calculators, very slow, small processors or to simulating the behavior of drums and disks. The price competition from drums and disks is strong. There is not as much demand for a read-only feature as semi-conductor people argue. The exclusively ROM memory will be forced to substitute for control logic or the storage of short microprograms. The ability to non-destructively read is desirable and would be exploited if Semi-Conductor memory were to achieve widespread application. The future of strewing small memories throughout a computer is not very bright as far as a market is concerned. Small memories replace registers, are used for paging and their use as caches is evolving. Beyond this the situation is chaotic. The example of a large computer system made up of the reiteration of numerous small computers such as is being pursued by the University of Illinois and Fairchild is not on the main stream of American computer development.

COST GOALS

The sales price of core memory systems being sold in 1970 is estimated in the following table:

SIZE OF SYSTEM	SALES PRICE PER BIT 1970
10,000 bits	5-6¢
100,000 bits	2-3¢
1,000,000 bits	1.5-2¢
10,000,000 bits	1.0-1.5¢

Price reductions by a factor of 3 have been made since 1965. This has been made possible by the availability of integrated microcircuits, reduction of core costs and simplification of stack design. It is not unreasonable to assume that the manufacturing cost goals in 1975 will be 40% of the amounts shown in the above table, or less.

SUMMARY

The argument given in 1967 for magnetics as the best choice for computer Main Memory in this same debate applies today.[1] "Integrated circuit memories, while finding use for small capacity stores, are not expected to replace magnetic memories in capacities as great as those needed for main memory (above 10^4 - 10^5 bits) primarily because of cost, manufacturing problems and reliability. We can probably expect a further increase in the types of memory approaches employed - another level in the heirarchy of memories. The system designers will probably have the final answer in determining the proper role and relationship of small, fast semi-conductor memories and the lower cost magnetic stores".

REFERENCE

1. R. J. Petschauer, "Magnetics - still the best choice for computer main memory", AFIPS Conference Proceedings, Vol. 31, pp. 598-600, Fall 1967.

RANDOM ACCESS SEMICONDUCTOR MEMORIES

David A. Hodges, Bell Telephone Laboratories, Incorporated

ABSTRACT

Semiconductor memories will become dominant in all fast (less than 1 microsecond) random-access applications by 1980. They will be superior to magnetic memories in cycle time, cost, physical size, power consumption, maintainability, and availability. The economic feasibility of intimately combining logic and memory functions opens important new doors for system designers.

INTRODUCTION

The choice of random-access store technology for future digital systems will be determined by systems designers' appraisal of the following parameters for the competing alternatives:

- System performance
- Total costs and trend of same
- System reliability and maintainability
- Device availability

This paper reviews the comparative characteristics of semiconductor and magnetic memories in these four areas of concern.

System Performance

Memory cycle time and access time are key parameters affecting the performance of digital systems. It has always been recognized that semiconductor memories can provide very short cycle and access times, equal to or shorter than those achievable with any other technology. Address decoding time and signal propagation delays will probably limit minimum access and cycle time for a kilobit store to the range of a few nanoseconds. Megabit stores with access and cycle times in the 50-100 ns range appear ultimately feasible. The high bit density helps to make these high speeds available at the CPU or other point of use of stored information, without serious limitations caused by wire delays. The natural NDRO character of semiconductor memory helps maximize system performance.

Semiconductor memory permits great flexibility in specifying system word length. Already at least one module has been offered which permits the user to adjust the word length to match specific systems requirements.[1]

Recently there have been several examples of semiconductor memory configurations in which the traditional flip-flop memory cell has been replaced with a simpler circuit arrangement, significantly reducing silicon area (cost) per bit.[2,3,4] So called "dynamic" circuit techniques are used; information is stored as charge on a capacitance and must be refreshed at intervals on the order of 1 - 10 ms. Cycle times in the range 0.1 - 1 µs appear feasible; the refresh cycle occupies only about 1 percent of real time. The system reliability of stores based on these designs is not yet proven; but this technique certainly will be useful in applications, such as data compression, where stored information is held only for microseconds.

Cost of Semiconductor Memory

Manufacturers of integrated circuits are delivering semiconductor memory in 1970 at prices ranging over various designs from 2¢ to 15¢ per bit.[5,6] Assuming that factory costs in optimum circumstances are 1/2 to 1/3 of selling price, this indicates expected 1970 factory costs ranging perhaps from 1¢ to 10¢ per bit. It seems certain that substantial reductions in the factory cost of semiconductor memory systems will be realized over the next five years.

Cost reductions will result from gradually automating processing, testing, packaging, and interconnection of integrated circuit components. Large savings will be realized by a shift to beam-lead of flip-chip packaging and interconnection. Smaller savings will be realized through continuing evolutionary improvement of all slice processing operations. Table I summarizes the technical and economic characteristics of two possible large-capacity store designs which should be manufacturable in volume not later than 1975.

These two hypothetical designs, which by no means define the full range of possibilities, give some indication of the variety of characteristics attainable. It is assumed that beam-lead or some functionally equivalent multichip packaging and interconnection technology is used. An example of an assembly of beam-lead chips is shown in Figure 1.[7] The low interconnection and packaging costs per chip result in a minimum overall cost when relatively small chips are used.[8] The relative cost for the two designs is roughly indicated by their relative total silicon area requirements for the complete store function. By 1975 the factory cost for the bipolar flip-flop design produced in volume should be in the range 0.25 to 1.0 cents per bit; that for the dynamic memory would be about 4 times lower.

Similar or identical circuit designs are applicable if Dual Inline Packages (DIP) or some similar package-per-chip approach is used. In this case, packaging costs per chip are substantially larger, and the minimum overall cost of a complete store module will be reached for chips approximately 2 to 3 times the area (with correspondingly more bits per chip) of those for the multichip approach.

Considering the lower slice processing costs and higher packaging costs for the standard package approach, it appears that the total cost per bit for complete store modules employing DIP or similar packaging will be roughly twice that for multichip modules. This estimate makes no allowance for the substantial capital costs and development expenses required to achieve low cost, high reliability multichip assembly techniques. These latter costs are of course a diminishing consideration when amortized over sufficiently large production runs.

System Reliability and Maintainability

Reliability has been one of the big questions about semiconductor memory. Field data is becoming available to indicate that 10^8 hours or more mean time to failure for interconnected integrated-circuit chips can be assured.[9,10] This translates to a mean time to device failure of 10^3 to 10^4 hours for a 10^7 bit semiconductor memory integrated at the level of 100 to 1000 bits per chip.

Store reliability appreciably greater than actual device reliability is possible through the use of error-correcting codes. At essentially no penalty in cost or performance, large semiconductor memories may be organized so that any single physical failure outside the common store control circuits causes only a single-bit error in the data word. Such a form of organization is uneconomic for most magnetic memory technologies. A Hamming code to correct single-bit errors and detect two-bit errors requires eight bits for a 64-bit data word. At least an order of magnitude improvement in store reliability is attainable by this approach, which has been used to advantage in existing commercial systems.[11,12] With error correction, a 10^7 bit semiconductor memory should show a mean time to failure (erronious data output) of 10^4 hours or more, comparable with good magnetic memories of similar capacity.

Store volatility is another aspect of the system reliability problem. Information is lost from a semiconductor memory when the power supply is interrupted. The solution to this problem is a back-up power supply or a nonvolatile back-up store. Highly reliable power sources, combining conventional power supplies with batteries and motor-generators, can be provided for large systems at a total first cost in the range of $4 per watt, or 40 millicents per bit for average operating power of 0.1 mw per bit. Back-up storage using disc files or magnetic tape can be provided for 1 to 50 millicents per bit. Key control information may be stored permanently in a read-only memory. It is interesting to note that most existing large systems employing nominally nonvolatile ferrite core memories already use one or more of these strategies for critical information. Thus it seems that the volatility of information in a semiconductor memory is a minor or negligible problem in large ground-based systems.

The above discussion indicates that the overall system reliability may be comparable for magnetic and semiconductor memories of comparable storage capacities. Consider now the question of maintainability. Semiconductor memories reach minimum cost for modules of 10^4 to 10^5 bits storage capacity; such a module is typically realized as a single pluggable unit. A large (e.g. 10^6 - 10^8 bit) memory would be built up of a large number of identical modules plus a single common store control unit. Such modular construction vastly simplifies the problems of diagnosis, repair, stocking of spare parts, etc. Because magnetic memories reach minimum cost only for modules of 5×10^5 bits or more, the advantages of modular construction are attainable only at cost penalty.

Availability

Now that early designs of semiconductor memory are being delivered, some of the earlier skepticism about the feasibility of this technology has been overcome.

However, there is still room for questions as to when semiconductor memories might be available and used in sufficient volume to challenge the present dominance of core memories. This could happen within two years if a major computer manufacturer elected to begin substituting semiconductor memory for core memory in continuing production of existing designs. Such a development is most likely in the case of computer makers with substantial in-house integrated circuit manufacturing capabilities. Improved system maintainability, as discussed above, would be the most likely rationale for a switch to semiconductor memory.

The fact that silicon integrated circuit technology is being actively pursued by many organizations for digital applications other than memory means that semiconductor memory development programs need not bear the full costs of materials and process technology development. This situation contrasts with that for new magnetic memory technologies, where development costs and schedule uncertainties seem likely to be greater. Finally, the availability of semiconductor memory will be enhanced by a continuation of the practice, already common for integrated logic circuits, of multiple-sourcing popular products.

Conclusion

The above discussion has attempted to show how and why semiconductor memories will match or surpass magnetic memories in all characteristics relevant to large random-access applications within two to five years. If these projections prove correct, they will (a) lead to significantly improved cost/performance characteristics for digital systems, and (b) cause major dislocations and realignments among components suppliers.

Acknowledgment

Helpful discussions with E. J. Alexander and T. R. Finch are gratefully acknowledged.

REFERENCES

1. Hart, T. W., Hillis, D. W., Marley, J., Lutz, R. C., and Hoffman, C. R., "A Main Frame Semiconductor Memory for Fourth Generation Computers," AFIPS Conference Proceedings, Vol. 35, 479-488, (1969).
2. Regitz, W. M. and Karp, J., "A Three-Transistor-Cell, 1024-Bit, 500 NS MOS RAM," 1970 ISSCC Digest of Technical Papers, 42-45.
3. Waaben, S., "Electronically-Variable Semiconductor Memory Using Two Diodes per Memory Cell," 1970 ISSCC Digest of Technical Papers, 46-47.
4. Boysel, L., Chan, W., and Faith, J., "Random-Access MOS Memory Packs More Bits to the Chip," Electronics, 109-115, February 15, 1970.
5. Gold, M. and Moran, J., "lc Bit Coal Prods Confab Debate," Electronic News, 1, February 23, 1970.
6. Fields, S. W., "Semiconductor Memories at a Glance," Electronics, 143-146, March 2, 1970.
7. Lynes, D. J. and Hodges, D. A., "A Diode-Coupled Bipolar Transistor Memory Cell," 1970 ISSCC Digest of Technical Papers, 44, 45, 182.
8. Wuorinen, J. H. and Hodges, D. A., "Component Technologies for Future Systems," IFIP Congress 68, Edinburgh, D59-D63, August 1968.
9. Platz, E. F., "Solid Logic Technology Computer Circuits - Billion Hour Reliability Data," Microelectronics and Reliability, Vol. 8, 55-59 (1969).
10. Peck, D. S., "Reliability of Beam-Lead Sealed-Junction Devices," Annual Joint Symposium on Reliability, Chicago, January 21-23, 1969.
11. Downing, R. W., Nowak, J. S., and Tuomenoksa, L. S., "No. 1 ESS Maintenance Plan," BSTJ, 1980, September 1964.
12. IBM System/360 Model 85 Functional Characteristics, IBM File No. 5360-01, Form A22-6916-0, (1967).

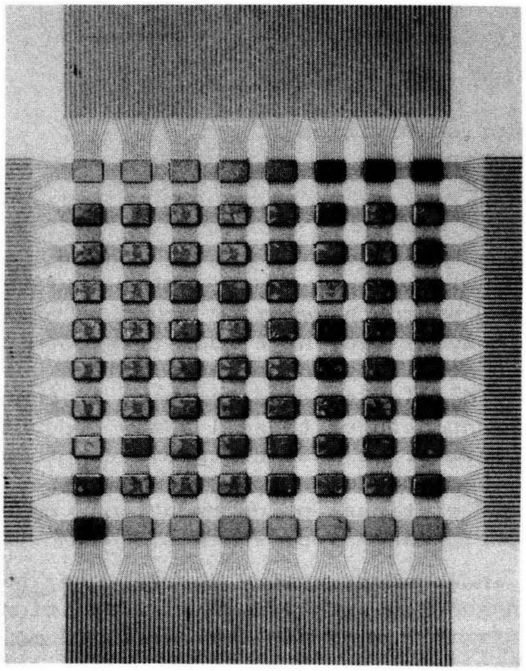


Figure 1

4096 bit memory array assembled on a 1.5 inch by 2 inch metallized alumina substrate.[7]

	Bipolar flip-flop Memory[7]	Dynamic Memory[2,3,4]
Module cycle time, ns	60	600
Bits/chip	128	512
Total Si area/bit, mil^2	44	12
Operating power/bit, mW	0.3	.05
Standby power/bit, mW	0.1	.03
Module capacity, bits	16,384	65,536
Chips/module (approx)	160	160
Module size, inches (approx)	3 x 4 x .25	3 x 4 x .25

Table I

Possible 1975 Semiconductor Memory Modules

CORE MEMORIES

P. A. Harding, Electronic Memories Division
Electronic Memories and Magnetics
Hawthorne, California

ABSTRACT

The Core Memory systems have been and shall remain the dominant technique for directly addressable data storage for at least the next 2 generations of main frame systems. The dominance can be justified by the many inherent advantages. In spite of its long tenure of 15 years, the core industry has hardly exploited the possible improvements in system organization, cost reductions, and batch processing. This paper shall illustrate the core market of the 70's and the techniques the industry will employ to obtain its objectives.

The random addressable memory systems users have watched the battle of the different technologies for the past decade in the anticipation of new breakthroughs in cost and performance. The dominant core has been challenged by ferite sheets, flutes, waffles, variable twistors, inverse twistors, laminates, chains, flat films, coupled films, ovonics, biaxes, cryogenics, plated wires and most recently the semiconductor memories. In every instance, the contending technology claimed either a new feature or a novel fabrication process as a panacea to the digital control memory. Some devices introduced extremely high speeds, others non-destructive readouts. Most concentrated on some form of batch or automatic process. The outcry was that the improved memory performance or new characteristic would be joyously accepted by the industries in order to increase the processing capabilities of the digital machines. Once accepted by the industry the new memory could be tooled for a high volume production and eventually be manufactured at lower costs than the core systems.

Unfortunately the market place has not accepted this congenial position. It demands low costs today. High performance or new approach memories would be accepted but only if they have near term potential of economically competing with the core. It is difficult for a computer or controller manufacturer to sell a new product at a higher price than his previous products. The electronics products consumer has demanded continual price reductions as well as performance improvements.

The independent memory manufacturer has been saddled with a number of additional constraints that has inhibited new memory technologies as well. Wide varieties of applications command memories that are economic over broad ranges in:

a) Capacity (10,000 to 10,000,000 bits)
b) Speed (0.5 μsec to 5 μsec)
c) Word Length (8 bits to 72 bits)

This flexibility, easily met by the core memories, has been a formidable obstacle to its opponents. The large development expenses necessary to design all the products for a broad market base cannot be tolerated by independent memory manufactures; the memory technology they employ must be naturally flexible.

Another miscalculation, the proponents of the competing memories have made is to under estimate the performance and cost potential of the core. The core memory has increased in speed, decreased in physical volume, reduced its power consumption and decreased in cost at a faster rate than should normally be expected of a mature industry; all this has been accomplished without a very sophisticated technology.

For these reasons the core dominates the present market. An estimate of the number bits delivered in 1969 are:

TYPE	ESTIMATED 1969 DELIVERIES	PORTION
3D-3 Wire Core	21 billion bits	56%
2 1/2D-3 Wire Core	11 billion bits	29%
Plated Wire	2 billion bits	6%
3D-4 Wire Core	2 billion bits	6%
2 1/2D-2 Wire Core	1 billion bits	3%
Miscellaneous	.5 billion bits	1%
	37.5 billion bits	

Almost 95% of the 1969 deliveries are core type systems. These systems vary in speed from the .50 to .75 μsec range (33% of the deliveries) to 4 μsec and slower (5% of the deliveries). At the present the increasing number of sales in the 500 nsec speeds indicate that the high speed core prices have dropped to an acceptable level. These speeds are utilized in wide range of sizes but the capacities have been following an upward trend and the average capacity for all computer systems installed in 1969 was close to one million bits.

Why has the core enjoyed this prominent position? The answer is simple; it has met the requirements of the industry but has been unsatisfied with status quo. Competition between various manufactures has forced a continual product improvement. As a result, todays core technology has many advantages.

Built in Decoding

The core threshold characteristic allows the drive economy of coincident current address selection and the address selection by means of current direction (phasing).

High Packing Density

The standard array density in present designs allows over 2500 cores/square inch. The low profile of the core and the frame upon which the array rests allows core planes to be on 1/8 inch centers.

Configuration Versatility

The core memory has many possible organizations, almost all are in production. For large systems the low array cost of 2 wire 2 1/2D memories can be used. For extremely high speed systems 2 wire 2D memories are available. For small systems where the drive circuitry must be economized the 3 wire 3D systems are best. Compromises allowing high speeds and large capacities are met by the popular 3 wire 2 1/2D memories.

Speed Versatility

The cost of memory systems are directly related to the speed of the memory device; not the system cycle time. High speed memory devices require fast currents or voltages, large power consumption and expensive mechanical and wiring structures. Fortunately the core can be manufactured in a wide variety of sizes and speeds such that the memory engineers can choose the appropriate core to match the system specification. Typical core manufacturers have available stock in sizes from 18 mil outer diameter (150 nsec switching time) to 80 mil outer diameter (3 μsec switching time).

Batch Processing

Inspite of its discrete component appearance the core memory fabrication procedures have many attributes of a batch processing system. In reality batch processing should be defined by the number of bits that can be handled per human operation. In this regard the core is superb.

FABRICATION STEP	OPERATION	CORES/ OPERATION
1	Mix Powder)	10^7 to 10^8
2	Roll Mill)	
3	Calcine)	
4	Roll Mill)	
5	Spray Drying)	
6	Sifting	
7	Pressing	32
8	Firing	50,000
9	Stringing	100
10	Terminating	50
11	Testing	1

The core achieves a high degree of batch processing with one additional advantage over all other integrated memory technologies; individual bad cores can be removed and the memory plan reworked.

Volume Device

The core allows microscopic imperfections without impairing its performance because it is a bulk switching device.

Established Industry

Because core manufacturers can design and deliver quantity production units in short order to meet specific customer requirements and because the industry has required more bits every year, the demand for this memory has continually risen.

This has entrenched the core technology; most companies have existing plant facilities with new requirements for immediate expansion. The expanding plant facilities have introduced improved tooling, improved designs and a continuous cost reduction of 29% for every doubling of cumulative production.

Improving Technology

Inspite of all the advantages discussed the core still has a long way to go. Direct extrapolation of the speed, capacity, cost and physical size progress curves indicate significant product improvements. Some impending developments in core memories are:

1. Extended core memory systems in the 10 to 20 million bit capacity range, operating at 1.0 to 1.5 microseconds cycle time and selling at less than 1.5 cent/bit.
2. The combination of very high speed semiconductor buffer storage systems (500,000 bit capacities) with extended core memory backing systems (100,000,000 bits capacity) will become popular because of the simplicity of interface between logic and memory; this combination will also provide rapid access to large volumes of stored data. Such systems should appear in a few years selling for less than 1 cent/bit.
3. Core memories will compete with discs and drums within the next five years. Minimum capacities of 100 million bits at less than 0.5 cent/bit should appear.
4. 2 1/2D core systems operating at less than 300 nanoseconds in one million bit configurations has been demonstrated in at least two laboratories and development of comercially available systems may be underway. If the market existed, these memories could be available in early 1972 at prices not significantly greater than todays' 650 nanosecond prices.
5. Extremely fast core systems (less than 150 nanoseconds) are possible by utilization of 2D or 2 core per bit techniques. The major problems associated with such designs are related more to the circuitry and hardware than to the magnetics. The decline of fast semiconductor costs and assembly costs will allow introduction of such systems by mid 70's.

All these improvements in price and performance can be accomplished by a continuous extension of the existing art.

1. Smaller cores - 14 mil outer diameter
2. Higher speeds - 100 nanosecond switching time.
3. Increased packing density - 10 mil centers and 10,000 bits/square inch.
4. Improved tooling and fixturing.
5. Greater utilization of lower cost 2 wire arrays.
6. Batch terminations.
7. Automatic array stringing.
8. A greater utilization 2D types of organizations.

Most of the stated improvements have been either demonstrated in a laboratory or are already in production in limited quantities.

These are not scientists dreams, but business men's practical evaluations.

RANDOM ACCESS MEMORIES - PLATED WIRE OR FERRITE CORE?

G. A. Fedde, Sperry Rand Corporation, Univac Division, Philadelphia, Pennsylvania

Ferrite core memories still have a market monopoly. In 1969, an estimated 2% of the 20 to 25 billion bits of magnetic random access computer main memory sold in the United States was plated wire.[1] The introduction of high performance plated wire memories in new products to be announced in the near future by major computer manufacturers will cause this percentage to grow fairly rapidly. Plated wire has the greatest advantages relative to ferrite cores in applications requiring less than 500 ns. cycle time. It is in these areas that the initial growth of plated wire will be concentrated.

Main memories, that will be used in new computer systems to be introduced in the next several years, will have cycle times of 1/2 to 1/10 microsecond. This is an approximate five to one reduction of cycle time compared to systems in volume production today. In this speed range the main competition for plated wire is expected to be semiconductor memories rather than ferrite cores. The modular capacity of plated wire memory will be 1×10^5 to 1×10^6 bits, about the same as today's production memory systems.

Extended main memory applications probably will be the area of direct price competition between ferrite cores and plated wire memories. Cycle times will range from 1/2 to 2 microseconds and modular capacities will be 8 to 16 million bits. Prices may be 1/2 to 1 cent per bit in the 1972-1975 period.

Technology projections can indicate potential opportunities and problems; they cannot prove whether plated wire or ferrite cores will provide the lowest cost memory systems in the 5 to 10 year future. Only actual production experience and evolving improvements can determine the issue. In the close competition between plated wires and ferrite cores, the economic result may be decided by non-technical factors. Timely management decisions and risk taking, vigorous and imaginative approaches to manufacturing and marketing are examples of such factors.

The market shares for plated wire and ferrite core will be determined primarily by the cost effectiveness and reliability with which the competing technologies can perform the desired function. All of the characteristics in Table I are certainly desirable and probably necessary for either magnetic memory technology to remain cost competitive in the 5 to 10 year future.

TABLE I

Essential Characteristics for Cost Competitive Magnetic Memory Systems

	Ferrite Core	Plated Wire
Proven reliability	Yes	Yes
2500 bit/sq. in. or more	Yes	Yes (In the laboratory)
Drive currents and voltages IC compatible	Marginal	Marginal
Power consumption less than 1 mw./bit	0.5 to 1.5 mw./bit	0.01 to 1 mw./bit
100 ns. cycle time - one element per bit	No	Yes (In the laboratory)
Single composition for wide temperature range	No	Yes
Multiple manufacturers, high quality products	Yes	Yes
Automated manufacture, assembly & test	Yes	Yes

Neither plated wires nor ferrite cores adequately satisfy the economical integrated circuit compatibility requirement today. The plated wire bit current, 40 to 50 ma. for 5-mil diameter wire, will drop to 15 to 25 ma. as 2-2.5 mil diameter wire is introduced. The plated wire word drive current, 800 to 1000 ma.-turns today, can be reduced to 250 to 350 ma.-turns by several means. The length of the bit will be reduced from 50 to 60 mils to 20 to 25 mils. Magnetic keepers, very closely spaced to the plated wire, are essential for the drive current reduction as well as proper operation of the closely spaced bits. Smaller diameter plated wires on 10-15 mil centers can help keep the required word drive voltage to a moderate value. Ferrite core drive currents can, of course, be reduced by making smaller cores and changing composition to reduce H_0. Smaller cores increase the cost of pressing, testing, wiring, and sensing, offsetting the advantage of reduced drive.[2] Reducing H_0 will increase the switching time for a given switching constant.

Ferrite core memories have a decided advantage in bit packing density compared to production plated wire memories. Ferrite core mats achieve 2500 bits per inch square relatively easily, while production plated wire memory planes are 550 to 1000 bits/sq. in. The closely spaced word drive lines and conformable keepers used to achieve integrated circuit compatibility will make it possible for plated wire bits to be packed 4000 to 5000 per sq. inch. Laboratory samples of plated wire planes now operate at 5000 bits/sq. in.

Plated wire memories exhibit a temperature coefficient of approximately -0.07% to -0.1% per degree centigrade. This is substantially less than the temperature coefficient of standard ferrite cores. Even the wide temperature range lithium cores have a larger coefficient of approximately -0.1% to -0.18% per degree centigrade.[3] A single composition element for all applications should give a cost advantage to plated wire by greater standardization and larger production volume.

The 10 megahertz operating rate using only a single element per bit is an obvious cost advantage for plated wires in those applications requiring the highest speed.

In the past, commercial magnetic memories have not been designed to minimize power consumption. However, power consumption will become increasingly important as memory systems prices continue to decrease and storage capacity increases. It is estimated that the lifetime cost of providing power to the system and removing the heat from the system and from the room is approximately $2 to $10 per watt.[4] Picking the average of $6 per watt gives a power consumption cost of 0.6 cents per milliwatt. If, as seems likely, extended main memory prices drop to 0.5 cents per bit, then 1 milliwatt per bit raises the total cost to twice the acquisition cost. Two milliwatts per bit triples the acquisition cost. Plated wire and ferrite core memory systems, designed 12 to 16 months ago with 600 ns. cycle time and approximately 1.5×10^5 to 2×10^5 bits, have power consumptions of 0.9 mw./bit and 4.8 mw./bit respectively. Current designs show a much larger relative power advantage for plated wire. A 2.5 microsecond 10^7 bit plated wire memory system requires 8×10^{-3} mw./bit while a 1.5 microsecond 4.7×10^6 bit ferrite core system requires 5×10^{-1} mw./bit. Multiplying the plated wire power consumption by 1.7 to adjust for the speed difference and 1.4 to adjust the capacity difference gives 2×10^{-2} mw./bit, 1/25 the power consumption of the core memory.

Ferrite core memories score high marks on the characteristics of reliability, automated manufacture and test, and multiple sources with quality products. Univac plated wire experience indicates that the plated wire element is as reliable as the ferrite core. The confidence factor is necessarily a little lower for plated wire because of the many times fewer element hours of operation. The degree of automation in plated wire memory manufacture is comparable to that of ferrite cores. There are at least five manufacturers with quality plated wire products available today.

Plated wire memories have the potential for gaining a majority of the non-IBM magnetic memory market. The most important characteristics in Table I, after proven reliability, all relate directly to low cost for whatever performance is needed. The better potential for plated wire in each of these characteristics can be the winning combination for memory users and suppliers.

REFERENCES

1. Mathias, J.S. and Fedde, G.A., "Plated Wire Technology - A Critical Review," IEEE Transactions on Magnetics, MAG-5, No. 4, 1969.

2. Russell, L.A., Whalen, R.M., Leilich, H.O., "Ferrite Memory Systems," IEEE Transactions on Magnetics, MAG-4, No. 2, 1968.

3. Greifer, A.P., "Ferrite Memory Materials," IEEE Transactions on Magnetics, MAG-5, No. 4, 1969.

4. Hodges, D.A., Lynes, D.J., "A Diode Coupled Bipolar Transistor Memory Cell," Proc. 1970 Int'l. Solid State Circuits Conf. (Oral presentation).

MULTIPLE TECHNOLOGY HYBRID SEMICONDUCTOR MAINFRAME MEMORY

Wally Raisanen
Motorola Inc., SPD
Integrated Circuit Center

In this short paper I will describe the properties and worth of multi-technology hybrid semiconductor structures as applied to mainframe memories for the decade of the 70's. I will describe the technical advantages of the hybrid structure and show how it results in minimum cost and maximum performance memory systems. System reliability and ease of system application are enhanced with the hybrid structure.

The essence of a hybrid semiconductor memory is a multi-chip assembly containing 4K to 16K bits of storage and all associated address translators, level changers, sense amplifiers, and drivers. It is generally a completely self-contained memory system with a logic level interface and a minimum number of extraneous timing signals. In contrast to the usual semiconductor integrated circuit situation, a hybrid structure offers the following advantages:

One may use any available solid state technology in any required combination to perform the three basic functions of a memory:

a. Data Storage
b. Address/Decode and Word Drive
c. Digit Sense and Drive

The electrical requirements of these three functions are quire different as they impinge on the solid state technology.

Data storage functions require minimum size and minimum processing complexity. These are very important considerations, as the cost of most semiconductor memories is completely dominated by the cost of manufacture of the storage array. It is well established that the MOS process technology provides lowest cost integrated circuit functions. MOS is cheaper than bipolar by a factor of 2 to 10 depending upon the application. Properly driven and sensed MOS storage arrays suffer no speed loss compared to bipolar structure of similar power dissipation.

Address decoding on the other hand is unquestionably best accomplished using saturated bipolar circuitry. Similarly, efficient word drive is best effected by bipolar transistors since this is primarily a current source or sinking application. MOS address and word drivers are 3 to 5 times slower than their equivalent bipolar structure for the same processing cost.

High speed, high efficiency digit sense is best done with bipolar amplifiers due to their stable and controllable threshold properties and high gain. Non-saturated non-gold doped process technology serves best in this function. Digit drive on the other hand requires saturated bipolar circuitry similar to that of the word drive application.

A hybrid structure allows each element of the memory to be separately optimized for cost performance. As shown above, the electrical requirements vary over a wide range and it is impossible to combine all these functions in one structure using one process technology without severely compromising some of the end objectives. These various memory structures can be divided into separate chips in a hybrid assembly at no cost penalty, thus allowing one to optimize the cost and performance separately.

Since even minimum-sized transistors can easily service up to 500 storage locations in the word and bit directions, it is impossible to obtain full efficiency of utilization from mono-lithic memory structures where address decoding and sensing must be combined with the storage array on one chip. To utilize the full capability of the peripheral circuits, a 250K bit storage chip would be required; an obvious impossibility with today's technology.

The one-technology approach inevitably compromises either the storage function or the interface levels. Since the storage function dominates the cost of the structure it is usually necessary to drive such devices with non-standard levels or codes. The hybrid technology allows direct optimization of interface levels at the module pins, saving time and cost in system applications.

Finally, in those areas where high reliability requires hermetically-sealed packages for the chips, multi-chip assemblies reduce the total number of hermetic sealed devices required by 6 to 9 times. Since the cost of processing the interconnecting part, be it beam lead or solder bump, is about 50¢ per square inch; it is clear that a net cost savings results as long as hermetic packages cost more than 10¢ each. Current prices range from 60¢ to $2 with projections in the 40¢ range for the far future. It is evident then that multi-chip assemblies will result in lower cost packaging for some time to come.

It is well known that the major failure mode in I/C's is breakage of the flying wire bonds. All practical hybrid multi-chip assemblies use beam leads or solder bumps for interconnections, thus vastly enhancing system reliability. Published results (1) indicate a reliability enhancement of two orders of magnitude for solder bumps over flying lead wire bonds.

A recent development (2) allows beam lead hybrid interconnection to be applied to I/C chips with no modification in the way chips are processed. The beam leads are formed as a part of the interconnecting wiring, which is fabricated using a two-sided "printed circuit" board process. This innovation allows hybrid structures to be manufactured for a cost lower than the cost of assembly of the chips into packages, substantially eliminating the cost of interconnection as a system cost.

In summary, the worth of multi-technology approach is as follows:

- Performance better than any single technology approach.
- Cost lower.
- Lower system power in large memories (mainframe) due to efficient utilization of peripheral circuits.
- Lower system design cost with fewer problems.
- Minimum number of man-made joints in systems for higher reliability.

---

(1) Norris and Landzberg. "Reliability of Controlled Collapse Interconnection." IBM Journal, Vol. 13 No. 3, May 1969, pp.266-272.

(2) Marley and Trolsen. "New Beam-Lead Connection Method Boosts Semiconductor Memory Yields." Electronics Vol. 42, No. 26, December 22, 1969, pp.105-110.

IN DEFENSE OF THE SINGLE TECHNOLOGY APPROACH TO SEMICONDUCTOR MAIN FRAME MEMORIES FOR THE '70's

Gordon E. Moore, Intel Corporation

Given that integrated circuit semiconductor memories will dominate the computer main frame in the 1970's, questions of the specific realizations that will be important becomes appropriate. In particular, will these memories emphasize multichip packaging techniques that allow the various functions (i.e., storage, decoding, sensing) to be realized with different semiconductor structures, or will compromises be made in order to include all the required functions within a single technology?

The case for the single technology approach is impressive.

Probably the strongest single argument that exists for the single technology realization is that by this means the total number of off-chip interconnections in the memory system is minimized. Such interconnections not only make a major contribution to failure rate, but contribute a significant portion of the cost to integrated circuits today. Their minimization thus is a long step toward lower initial cost and improved reliability of operation -- probably the two most important measures of merit for a main frame memory system, once adequate speed is achieved.

An example of the magnitude of this reduction is instructive. Consider a "plane" of semiconductor memory 4096 words by one bit composed of 1024-bit integrated structures, in one case with full decoding and sensing on each of the chips and with a separate decoder and sense amplifier in the second case. In the first case, 16 leads per chip is adequate (ten addresses, one data, one read/write control, three power, one for good luck), or 64 total off-chip connections for 4096 bits. When constructed of undecoded 1024 bit arrays and separate decoding and sensing, each memory array chip requires at least 66 leads with an additional 40 or so for each the X and the Y decoders. The sense amplifier is additional. Thus, at least five times as many connections are required for separate chips.

On the other side, incorporation of the peripheral circuitry on the chip results in duplicating some functions several times. In the example above, the decoders were duplicated four times in one case, where if they were equipped with proper drivers, they could actually be shared by several planes in the second case. This is not such a great disadvantage, however. By duplicating the circuitry, the drive requirements of one portion of the decoder are minimized. In effect the drivers can be distributed throughout the system, rather than concentrated in one device. In this manner the difficult-to-integrate, high-current driver devices are spread into several easy-to-integrate, small structures. By duplication, chip area -- a principal measure of cost of integrated circuit -- can often be decreased. Sense amplifier circuitry can be duplicated several times in the area it would take to make lead-bonding pads for a similar amplifier on a separate chip

The next question to address is, Can adequate performance be obtained with the single technology? The answer is a strong affirmative. In fact, the highest performance can be obtained by utilizing all-bipolar circuitry. System access times as low as 15 μsec have been quoted utilizing an all ECL circuit approach. A 500,000 bit main frame memory with cycle times less than 200 μsec utilizing all-bipolar circuits with decoding and sensing on the chip has been described.[1] Even with all-MOS approaches, the performance can be made consistent with main frame requirements. An all-MOS, decoded-on-the-chip structure useful at 500 μsec cycle time in large memory systems was the subject of a recent paper.[2] Thus, the performance is already at least competitive with standard main frame techniques.

Unless the multi-technology approaches offer advantages of cost in some small portion of the performance domain, it is hard to see where they fit in. The region in which they potentially offer this possibility is certainly limited to cycle times below 300 μsec, if it exists at all, and is shrinking rapidly as the all-MOS system performance improves.

An additional advantage to the single-technology, fully-decoded approach is the simple interface it allows between semiconductor manufacturer and user. Conventional integrated circuit packages such as 16-lead dual-in-line structures can contain 1024 bits and are useful as simple, flexible building blocks for system realization. The throw-away module is small and inexpensive. For the other approach, it is necessary either to go to large, many-leaded packages each containing one function, which are expensive and difficult to use; or it is necessary to go to a complex multichip package containing all the functions. The latter is not flexible and can be a relatively expensive throw away.

All aspects considered, it is my opinion that the single technology approach combining storage, decoding and sensing on the same semiconductor chip offers significant advantages in cost, reliability and flexibility. These advantages will assure that such an approach will dominate main frame memories in the '70's.

REFERENCES

1. F. S. Greene, Jr. and N. V. Phan, "Megabit Bipolar LSI Memory Systems: IV" Digest of Technical Papers, 1970 International Solid State Circuit Conference, Philadelphia, Pa., (Feb. 1970) p. 40

2. W. M. Regitz and J. A. Karp "A Three Transistor Cell, 1024-bit, 500 ns MOS RAM" Digest of Technical Papers, 1970 International Solid State Circuits Conference, Philadelphia, Pa., (Feb. 1970) p. 42

COMPUTER TERMINALS FOR NEWSPAPERS: APPLICATIONS AND REQUIREMENTS

Richard S. Marcus, Electronic Systems Laboratory, Massachusetts Institute of Technology
and
Jules S. Tewlow, American Newspaper Publishers' Association, Research Institute

ABSTRACT

Five areas in which interactive computer terminals may be required for news processing in the automated newspaper of the future are identified. These are (1) editing of news story texts; (2) classified advertising; (3) display advertising; (4) page make-up and layout; and (5) news story storage and retrieval. The requirements imposed on the terminals for these applications are shown to be beyond present capabilities in some cases.

AUTOMATION OF NEWS PROCESSING

Newspapers are becoming more automated. The line justification and hyphenation aspects of the typesetting operation are presently being performed by computers in a wide variety of locations. In a typical operation the Worcester Telegram uses a Digital Equipment Corporation PDP-8 to accept unjustified news story material prepared on punched paper tapes and to generate justified, hyphenated punched paper tapes for input to the linecasting machines. Scattered developments in other areas may also be cited to indicate the current trend. The Boston Globe has installed a system based on the Honeywell 1200 which, in addition to the typesetting function, has the feature of online keyboarding of the text material so that the preparation of input paper tapes is bypassed. The Washington Star has developed the feature of computer-assisted editing of news stories. The Miami Herald has already developed the feature of rapid computer credit-checking of classified ad customers as part of their operations on the IBM 360/30. The Los Angeles Times is developing a classified ad system in which the ads will be input to the computer in a display console where they will be subject to immediate correction by the typist. The New York Times is developing a storage and retrieval system for its clipping files based on an IBM 360/50. Catalog information stored in digital form and full text in microfilm form would both be accessible from display consoles. These, of course, are just examples and do not cover the range of applications even at the newspapers cited. Standard accounting and billing procedures, among others, would have to be added for completeness.

Looking farther in the future we can project a continuation of this trend toward automation. The process of publishing the news has been described[1] as centered on the information transfer among many individuals and machines in which quick reaction time to the rapidly changing news and advertising situations is essential. It is then natural to project that the central element of the news processing system of the future will be just that tool now being developed to expedite information processing and its interaction among man and machines, namely the time-shared digital computer. Some possible elements of this computer-based system have been previously projected[2] and are pictured in Fig. 1.

INTERACTIVE COMPUTER TERMINALS

The main theme of this paper is to consider the means by which the human will communicate with the computer-based news processing system, namely the computer console or interactive terminal. We shall concentrate on the requirements on the consoles so that certain news functions can best be performed. These functions are (1) news story editing; (2) classified ad processing; (3) display ad processing; (4) page makeup and layout; and (5) news story storage and retrieval. Of course, there are many other applications which are likely to be part of the newspaper operations and require consoles. For example, the publisher could wish to use a management information system -- from a console in his own office -- to see quickly recent circulation or advertising figures. In another situation the business analyst might want to scan and change a PERT-type chart in considering some new plant development. As a third example, a programmer on the newspaper staff might wish to apply an online debugging routing to correct a program. Because these kinds of situations may be considered as typical of future computer applications for any medium to large business organization and have been discussed elsewhere,[3-5] we shall emphasize only the five areas specifically related to newspapers listed above.

The specific nature of the applications and the consequent requirements for computer terminals depends heavily on the overall system configuration in which the news processing takes place. We shall assume, as we have above, that the news process is basically automated with a time-shared computer serving as the central component. To the extent that this assumption is not yet satisfied, the requirements given below may need revision for the present-day situation.

EDITING OF NEWS STORY TEXT

As the newspaper business evolves from low-speed typesetting by means of hot metal mechanical linecasters having a high error rate of one or two percent to high-speed, photo-mechanical and electronic typesetting devices whose error rate is measured in one per tens of thousands of characters produced, it becomes imperative that all proofreading and corrections be done before the fact -- rather than after the fact as now is the case.

The reason for this dramatic change in methodology is that in photo-mechanical and electronic typesetting, the correction of errors, typographical and otherwise, is a very time-consuming process which would and does impair the inherent efficiencies and labor economies derived from utilizing such sophisticated equipment. The easiest way to perform these proofreading/correction functions before the fact is by using display terminals. Thus, one can readily see that in a business which must meet several production and distribution deadlines daily -- the act of getting out the news -- that computer-related display terminals are the only devices which properly interface with current and future high-speed typesetting technology.*

Essentially the basic functions of an editorial terminal would be to display sections of a given story on file in computer-based mass storage and check the displayed text for factual and typographical correctness and style. Character, word, sentence, or paragraph additions, deletions, changes, and rearrangements would be performed on the display unit's typewriter keyboard.

The users of such a display terminal might include (1) a reporter generating a story originally; (2) a typist inputting the story into the computer from printed, written, or verbal form; (3) an editor reading the story for possible changes in content or length or just scanning stories in the data bank for general interest. The minimum character set requirements are about 90 characters: upper and lower case** alphabetics, the numerals, and the common punctuation marks and spacing.*** Another 10 to 20 characters would be handy for typesetting control, and, possibly, the regular ASCII (American Standard Code for Information Interchange) character set could be made to serve. It would be useful to have one or two additional fonts to represent italics and/or boldface but these features could be indicated with special marks rather than shown as such. Similarly, one size of type should be sufficient; the headline, if it is needed at all at this point, may have size indicated and need not appear full size on the display. Variable character widths and line justification on the right-hand margin are probably not needed either.

An important functional requirement for editing is a means of indicating that portion of the text to be modified. The light pen is perhaps the best tool for this purpose. Another device which can be used for this purpose is the cursor operated by special controls (for example, a "joystick" or electronic tablet or "directional" buttons). In the optimum configuration a set of cursors might be used to delimit various text strings while the light pen is used to point to strings or locations. A third kind of identification of text location is by context, that is by typing some portion of the text itself. Of course, any of the three locating mechanisms may be used alone but a combination of all three should provide the most powerful capability.

Other special controls would be desirable to indicate which editing function -- insert, delete, etc. -- is being requested. Fixed-position keys separate from the regular keyboard would suffice for this purpose but the addition of programmable function buttons* would add capability and flexibility.

In the proofing mode a variable-speed roll (moving window) of the text in a standard-size column format seems desirable. An easy-to-operate page flipping mode would be useful in addition and could serve as a substitute for the roll feature as a second-best situation. While a 30-character line should suffice for a single column, the column-and-a-half and double-column formats would require up to 60 characters. A sufficient number of lines to provide adequate context is necessary. Perhaps about 20 is a minimum with about 60 being desirable if the screen is big enough for legible viewing.

The speed of presentation or of rolling the text should be capable of allowing for high proofing rates, at least in bursts. For this mode we may wish for up to 1500 words per minute or about 120 characters per second. A much more severe requirement is for the editor scanning stories in a "page-flipping" mode. Here about 5 pages a second might be desired or up to about 15,000 characters per second.

One feature that is important for a good editing capability is a rapid adjusting of the text display as changes are made. For example, on deletions, "holes" are filled up and any adjustments to the given line or succeeding lines for justification are

---

* OCR (Optical Character Reader) devices might also be of some use where offline preparation of input is more practical then online input.

** For a discussion of the requirements for both cases see Kinney.[6]

*** These figures are based on English, a typical alphabetical language. Languages like Chinese or Japanese, with as many as 10,000 characters, obviously present vastly more difficult problems.

---

* See, for example, Haring.[7]

automatically accomplished. On additions, lines, or portions of lines, may be squeezed off the bottom of the page but, of course, are retained in memory. In order to re-display the corrected page in a fraction of a second we must have very fast data transfer as for the editor scanning mode.

Immediate hard copy output may not be necessary where there are sufficient display terminals at which the current version of any story is always available with easy editing and fast scanning capabilities. For stories not having a near deadline, however, an offline, or deferred hard-copy, output might be desirable to allow an editor, for example, to review and reword a story at his convenience away from any terminal. It might be possible to use the regular newspaper production line, modified to make a single proof, for this purpose.

CLASSIFIED ADVERTISING

In most daily newspapers, the bulk of classified advertising arrives via the telephone. A classified-ad phone operator who takes the copy could, in typing out the ad, preview the text on the face of her display set. While the customer is still on the telephone, the ad taker could check with him for correctness of the information prior to releasing the copy to the computer for subsequent manipulation, etc. Likewise, the ad taker could receive back from the computer a credit check and price of the given ads before the customer hangs up.

In classified ad display it is still possible, as with news story text, to get by with one font and one size but it is much more desirable to be able to display these features exactly as they will ultimately appear in the paper, especially where font or size variation exists within a single ad. Since the basic unit is one ad, page size requirements are greatly reduced. A large classified ad might be 20 lines. Ads bigger or more varied than allowed for on the classified-ad terminal could be handled as display ads (see below).

Editing for the classified ad would be similar to story-text editing. Counterbalancing some simplification made possible because the ad is likely to fit whole on the screen is the complexity caused by multiple fonts.

Since the operator is taking the ad verbally from a customer, the required speed of display is much reduced from that of the text editing console. On the other hand, one problem with automation of the classified advertising subsystem is the unevenness of input flow. A moderate-size newspaper may require 50 or more telephone operators to handle the peak load on Friday for the Sunday edition. However, less than one-third that number may be required earlier in the week. Then too, there are peaks and valleys in demand from minute to minute during any day due to the stochastic nature of the advent of customer calls. Providing one telephone each for the maximum number of operators may not be too costly but a like number of online display terminals might easily "break the bank" because of underemployment.

One solution to the problem of unevenness of input flow that has been suggested (by the Los Angeles Times) is to continue to take the ads over the phone and then batch them into the computer over display terminals. This procedure loses out on the online feature but smooths out the stochastic nature of the inputs, especially if time-sharing with text-editing consoles is considered. Another solution, somewhat farther in the future, would have a fully automated classified ad taking facility. The customer would call up from a picture phone with a pushbutton dialset. The customer would key in the ad and see the resultant printed ad format. Both visual and oral instructions could be employed to help the customer. A human could monitor several incoming calls at once and be available for personal assistance.

DISPLAY ADVERTISING

A well-integrated, economical, computer-based display advertising subsystem is probably that component of the news processing system of the future that is the farthest away from reality of any of the five components discussed in this paper. The problem is the difficulty of digitizing artwork and photographs and integrating their handling with the associated text. Without trying to predict the dividing line or the underlying processing mechanisms too finely, we shall assume that some information is kept in analog form (say photographs and sophisticated artwork) and some is kept in digital form (say text and simple graphics).

The display ad terminal might then have a keyboard, controls, light pen and display screen as for the editing terminal, and with similar functions as for editing or classified advertising. There would also be an input station for the analog graphics and an electronic (Rand-type) tablet for tracing. The need for hard copy depends on three other interrelated matters: (1) the availability of the display screens at inhouse (newspaper) and customer locations (note that there may be several locations for one customer with decentralized operations); (2) the adequacy of the display to fully represent the ad; (3) the ease and speed of using the regular typesetting process to get proofs.

The newspaper considers it very important that the customer see the ad exactly as it will appear in the paper. Therefore, if the customer were to use a display screen for verification, it would have to accommodate gray-level photos and artwork as well as multi-font characters. To satisfy the requirements of ad composition (mark-up), proofing, and correction we can get by with lesser capabilities. The graphics could be limited to indicating size of the artwork and perhaps a few distinguishing features (rough outline) to aid in layout and possibly cropping. Here is where the electronic tablet could prove useful. A provisional layout as generated by the customer would be traced with the electronic stylus to give sizes and outlines. Similarly, limitations on font representation could be imposed but to the extent that they are the capability for effective composition without need for later reworking will be degraded. Here it is more important to represent sizes accurately than to exactly portray frilly fonts.

Newspaper ads can be as large as full page. It is not necessary that full detail be available for the full page display; full detail on sections of the ad may be sufficient.

Input of simple sketches via the light pen is certainly feasible. Just how far this can be carried and to what extent it would be superior to pencil sketches with analog representation -- ease of modification is one possible advantage -- is unclear.

PAGE MAKE-UP AND LAYOUT

Page make-up and layout through computer assistance is close at hand and can be accomplished fairly simply if the editing and display advertising terminals are available to the make-up man. The additional function that would be required would be the representation of a page as blocks (rectangles) tagged to indicate the story or ad contained in the block. Placing of blocks could be largely algorithmic and performed by the computer with manual override for computer goofs or general esthetic principles. The make-up operator would use a light pen to aid in rearranging.

In particular, the make-up display console will contain all the controls necessary for the make-up editor and display ad compositor to efficiently perform the following tasks:

- -- Provide access to all news stories, photos, reserve news, and/or display ads in storage.
- -- Display these materials in a convenient form.
- -- Permit modification of the input material -- e.g., trimming, filling or leading-out particular stories.
- -- Allocating each element into its proper location on the page in accordance with the page layout or dummy.
- -- Display all pictorial materials in some representative fashion and have the capability of moving and cropping these materials as required.
- -- Provide means for previewing the finished page.
- -- Create a record on a disk or magnetic tape of each page prior to final typesetting.

NEWS STORY STORAGE AND RETRIEVAL

The users of this subsystem would include reporters and editors and, possibly, the general public who could purchase information service using terminals located remotely from the newspaper. Details of the plans for a computer-based newspaper library retrieval system have been proposed elsewhere.[8] Two different kinds of terminals are required, though a recent study[9] has shown how they can be at least partially combined.

One kind of terminal is used to view the story itself. Because the story may contain photographs and other pictorial information, a gray-level capability is required for the display. As stated before, story scanning can require, in a page-flipping mode, up to 5 pages a second, although a page every second or two would still be satisfactory for most situations. Hard-copy output is a definite requirement, although the ready availability of display terminals can reduce this requirement. A page per second hard copy output would probably be sufficient. Immediate initiation of this output would be desirable but a delay of a minute or so would be tolerable.

The second kind of terminal is needed as a "finding tool" for the news articles. The user interrogates a computer-stored catalog to news stories by an interactive dialog conducted over this terminal. This same function is also useful for a news editor wanting to select interesting stories from the days budget of wire service and in-house generated stories. A fairly simple upper and lower case font requirement should suffice for news story selection. (Note that this requirement is much less severe than selection for technical and scientific literature where more complex symbology is found.) Pointing by light pen and/or cursor and other special functions would be useful as for the editing console. An editing function per se would be useful on a limited basis to prepare bibliographies. Similarly, a hard-copy output would be desirable for general review, especially to the extent that a very convenient -- and fast -- page flipping and rolling ability were not available.

OTHER CONSIDERATIONS

An estimate of the values and costs of these terminals would require a much greater in-depth analysis of news processing operations and means of implementation than we have been able to perform in this paper. However, it might be informative to give a notion of what the authors feel would be a reasonable price range for terminals which might be used economically for operations in present-day news processing: classified-ad taking (simple one-operator per customer terminal) -- $2, 500 - $3, 500; display-ad console -- $40, 000 to $75, 000, all others -- around $5, 000.

There are many basic questions of means of implementation left unanswered. These include: storage-tube versus refreshed displays; hard-wired logic versus programmable logic; input devices; and standalone versus buffered clusters. However, some fairly simple observations may be made. The requirements for rapid revision and redisplay in several of the terminals would seem to favor a refreshed display operating from a drum-type memory. Buffered clusters would seem to be desirable to reduce costs. The flexibility of programmable logic would seem favored until specific functions can be established firmly enough so that changes are infrequent. Human factor considerations, such as ease of viewability and screen adjustment, could make the difference between success and failure.

SUMMARY

Five applications specifically related to news processing have been identified in which interactive computer terminals may be useful. These are (1) editing of news story texts; (2) classified advertising; (3) display advertising; (4) page make-up and layout; and (5) news story storage and retrieval. The newspaper of the future could be a highly automated operation with humans acting in a controlling and monitoring capacity directing the five functions listed above, as well as many others, through interactive terminals connected to a central, time-shared digital computer. However, the requirements on these terminals to perform these functions have been shown to be beyond present capabilities in some cases. This may be due, in part, to the fact that some of the requirements may be more severe than previously supposed.

REFERENCES

1. Reintjes, J.F. and Marcus, R.S., "Computer-Aided Processing of the News," Proceedings of the Spring Joint Computer Conference, May 14, 1969. Vol. 34, pp. 331-338.

2. Tewlow, Jules S., "Time-Sharing and the Newspaper of Tomorrow," ANPA Research Institute Bulletin 951, April 15, 1968.

3. Hobbs, L.C., "Display Applications and Technology," Proceedings of the IEEE, Vol. 54, No. 12, pp. 1870-1884, December, 1966.

4. Siders, R.A., Computer Graphics , (American Management Association, New York, 1966.)

5. Licklider, J.C.R., "A Picture is Worth a Thousand Words -- and it Costs . . . ", Proceedings of the Spring Joint Computer Conference, May, 1969, Vol. 34, pp. 617-621.

6. Kinney, G.C., and Showman, D.J., "The Relative Legivility of Uppercase and Lowercase Typewritten Words," Information Display, Vol. 4, No. 5, September, 1967, pp. 34-39.

7. Haring, D.R., "A Display Console for an Experimental Computer-Based Augmented Library Catalog", Proceedings ACM National Conference, Las Vegas, Nevada, Aug. 27-29.

8. Computer-Aided Processing of the News, Second Status Report, Electronic Systems Laboratory, Report SR-396, August, 1969.

9. Haring, D.R., and Roberge, J.K., "A Combined Display for Computer-Generated Data and Scanned Photographic Images," Proceedings of the Spring Joint Computer Conference AFIPS, 1969, pp. 483-490.

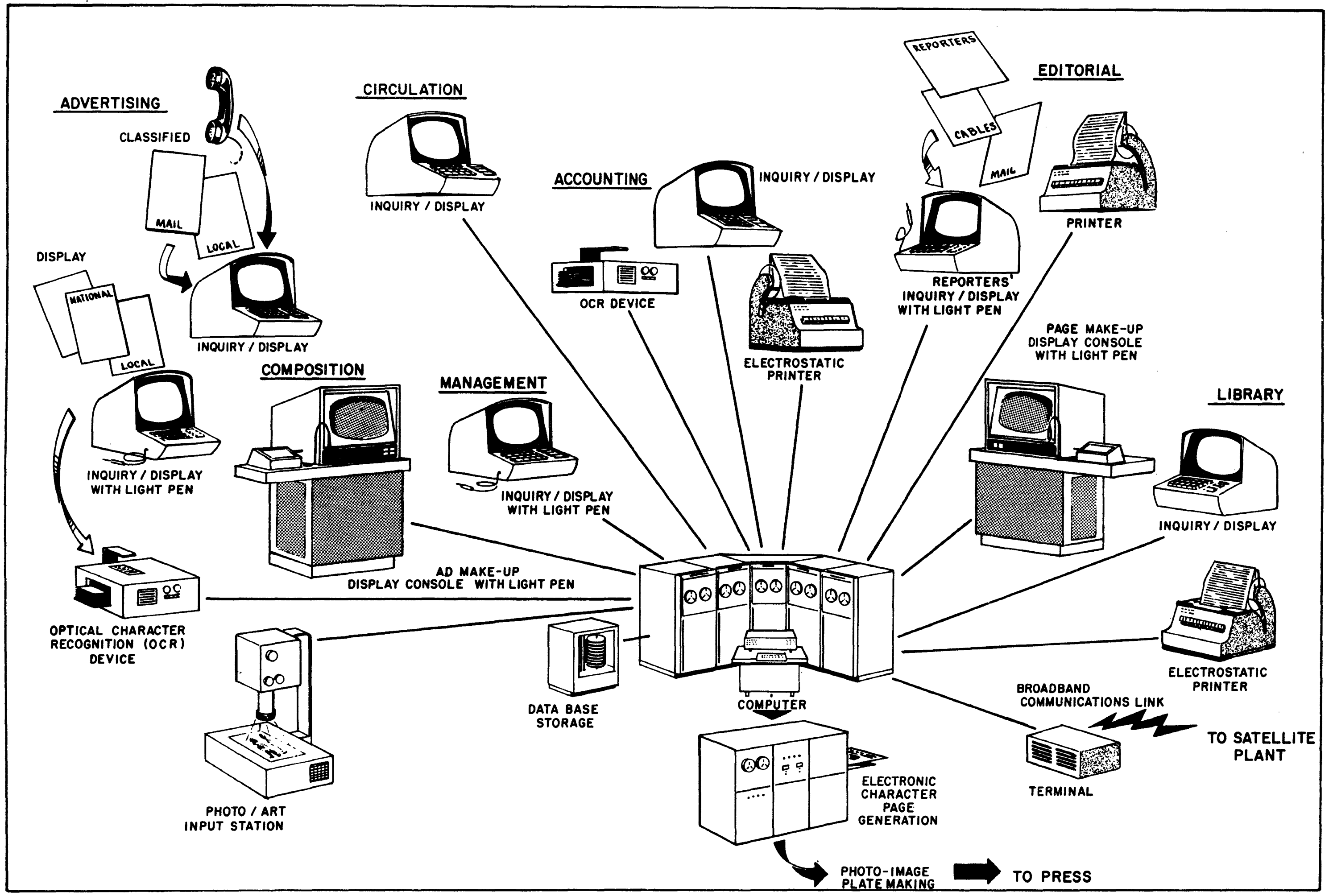

ADVERTISING
CLASSIFIED
MAIL
LOCAL
DISPLAY
NATIONAL
LOCAL
INQUIRY / DISPLAY
INQUIRY / DISPLAY WITH LIGHT PEN
OPTICAL CHARACTER RECOGNITION (OCR) DEVICE
CIRCULATION
INQUIRY / DISPLAY
ACCOUNTING
INQUIRY / DISPLAY
OCR DEVICE
ELECTROSTATIC PRINTER
REPORTERS
CABLES
MAIL
EDITORIAL
PRINTER
REPORTERS' INQUIRY / DISPLAY WITH LIGHT PEN
PAGE MAKE-UP DISPLAY CONSOLE WITH LIGHT PEN
LIBRARY
INQUIRY / DISPLAY
COMPOSITION
MANAGEMENT
INQUIRY / DISPLAY WITH LIGHT PEN
AD MAKE-UP DISPLAY CONSOLE WITH LIGHT PEN
PHOTO / ART INPUT STATION
DATA BASE STORAGE
COMPUTER
ELECTROSTATIC PRINTER
BROADBAND COMMUNICATIONS LINK
TERMINAL
TO SATELLITE PLANT
ELECTRONIC CHARACTER PAGE GENERATION
PHOTO-IMAGE PLATE MAKING
TO PRESS

Fig. 1. Possible Elements of the Future Newspaper System

THE WIRE SERVICE AND THE VIDEO DISPLAY

James F. Darr, Communications Department, United Press International

ABSTRACT

During the next two to three years there will be a dramatic change in the methods used by United Press International for gathering, editing and distributing its news reports to subscribers in the United States and other parts of the world.

There will be a transition from manual methods of editing and distribution to an automated system.

The editors' copy pencils will be replaced by keyboard operated Video Display Terminals, the subject matter of this paper, which is considered the key element of the automated news system.

INTRODUCTION

When Horace Greeley first exhorted that young (newspaper) man to go West, he was not aware that he was sending him to the future mecca for computer peripheral devices. Today, when a newspaper editor heeds that advice, he may very well be on his way to visit one of the many CRT manufacturers who are dotting the California countryside. The processing of news information is about to enter a new era. The computer has already found its way into the newspaper industry, and there is ample evidence that the newsmans' typewriter and copy pencil may soon be replaced by video display devices. Before this major step is taken, however, there are a number of problems to be solved - both human and technical.

The following pages will outline: (1) the technical requirements of the newspaper industry, and in particular of the news wire service organization, for display terminals; and (2) the approach taken by a major CRT vendor in the design and manufacture of displays for commercial computer applications.

The human problem areas mentioned above could certainly be the object of a separate, and voluminous, presentation. We will not attempt to analyze those problems here, but before beginning our technical evaluation, it is possible to spotlight one primary area of concern which is common to almost any endeavor; man's resistance to change. The opposing points of view in the news reporting industry can be summarized by two recent Editor and Publisher articles. The first explains the approach taken by the Southern Newspaper Publishers Association (SNPA) Foundation, whose first of ten 1970 industry seminars is entitled "Effects of the New Technology on News Reporting." Items to be discussed include:

- Effect of electronic data storage and retrieval systems, and decline of hard-copy systems.
- New skills that the reporter must acquire to do his job effectively.
- Effects of technological advances on news reporting methods.

The second article presents the sentiment of a great many reporting and editorial personnel throughout the industry. In discussing newsroom automation, an Associate Editor:

- expressed his "fear of computer companies and electronic whizzes;"
- declared that there is a "sense of the mystique" about editing the news, and only the editor, with his particular skills and experience, has it;
- stated that many editors have become "suspicious of bluesky production managers and equipment salesmen."

These opposing views present, in a nutshell, a primary human problem area. Before we replace the typewriter and editor's pencil, we must convince the newsman that our CRT-computer approach will provide him with a more efficient tool for editing or reporting the news, without robbing him of his editorial prerogatives.

An Electronic News Editing System

To begin our discussion of the technical requirements of the news wire service organization for display terminals, a brief description of the projected information handling system at United Press International is appropriate.

In an effort to tailor the news service to the needs of its newspaper clients, and to provide them with error-free copy, UPI has on its drawing boards a plan for a computerized system which will store incoming news stories from all over the world on random access devices for subsequent transmission to clients.

An abstract of each story will be automatically generated and delivered to editors in UPI's news room on teleprinters. The editors will scan the abstracts, call up the stories on keyboard-operated cathode ray display terminals, edit them on the CRT by making necessary corrections, insertions, deletions, substitutions, etc., and transmit the story back through the computer to random access storage for subsequent delivery to UPI's subscribers.

In the early stages of development, the electronically edited news stories would be delivered to subscribers over UPI's news service wires. Subsequently, in the foreseeable future, the abstract wire could be extended to newspapers' offices where their editors would select the stories from UPI's computer(s) using the national telephone network as a connecting communications facility. The news stories would be delivered at speeds in excess of 1,000 words per minute and would be recorded on hard copy printers and/or punched paper tape. Higher speed transmissions would be recorded on magnetic tape or even be made directly into a newspaper's own editorial computer system.

The system, then, calls for the use of the video display terminal for editing of copy, and for composing and rewriting of stories. In evaluating CRT devices in light of the specific UPI requirements, an "ideal" wire service video display terminal has been defined. The following is an outline of the capabilities which this ideal unit should possess:

(1) Both upper and lower case alphanumerics must be displayed. This is a firm requirement for editing of news stories.
(2) A standard typewriter keyboard should be the input device, to facilitate editing and story writing, and to minimize retraining. Function keys should be added to provide for certain editing functions such as "Flush-left", "Flush-right", "Center" "Bold-face", etc.
(3) Line length should be at least 80 characters, with a screen capacity of thirty lines. These figures relate both to present teletype formats and to average news wire story length. The CRT local storage capacity should accommodate at least two full screens so that at least two "pages" of copy can be handled without going back to the computer.
(4) The characters displayed should be steady, free of jitter, and easily readable under normal room lighting conditions. A unit that lacks good screen clarity could easily drive an editor back to his pencil and typewriter.
(5) Data insertion capability should be provided, along with the ability of the hardware to identify and display only full words at the end of the line. If the insertion of data in a line causes a portion of a word to be shifted to the beginning of the next line, then the entire word must be shifted. If insertion on a full screen causes characters to be lost (or dropped off the bottom), these characters should be automatically stored.
(6) A character delete capability must be provided, and displayed text should close up on the screen after a deletion. A simple closing of the line where the deletion occurred would be satisfactory.
(7) The video display should accommodate the shifting of a word, sentence, or paragraph from one part of a story to another. Words, phrases, sentences, and paragraphs could be moved from one area of the screen to another.
(8) A "character transpose" key should be available for the correction of this very common error. Positioning of a cursor under the first of two transposed characters, and depressing of the specific function key, would interchange the two characters.
(9) There should be provision for "rolling" copy up and down the screen in a smooth manner, with the speed of the "roll" being variable to suit the pace of the person editing the text. This feature could be controlled by a "roll-wheel" or "joystick" conveniently located adjacent to the keyboard, and would permit rolling through a complete story in either direction. This implies the ability to automatically retrieve text from the CPU, if the entire story exceeds the capacity of the display memory.
(10) Reliability of the terminal is a high-priority item. For this reason, stand-alone units with self-contained memory and control electronics,

are preferable. With multiple units driven by a single-controller, the loss of all units caused by the loss of a controller, would be unacceptable.

(11) The unit should be able to adapt to printers which will provide hard copy at speeds ranging from 150 baud for teleprinter, up to 2400 baud or higher for electrostatic printers.

(12) The terminal must be able to display selected information in a bold-face manner. Color, slanting, or increased brightness could be employed.

(13) Information should be transmitted between CRT screen and CPU at no less than 2000-3000 words per minute. This would permit filling or dumping of the screen in about six seconds. Faster operation would be preferable.

(14) The unit should have the capability of alerting an editor to a news flash. Flashing or colored characters on a reserved area of the screen could accomplish this.

(15) Hyphenated and justified text should be displayed in justified form with even right hand margins. Psuedo justification, using additional increments of inter-word spaces, would be satisfactory. The terminal will be used to correct inaccurate word hyphenations that sometimes occur in the process of justifying copy. The ideal terminal-CPU function would automatically rejustify and redisplay the same block of copy as hyphenation corrections are made. It is possible that a subsequent line in the same paragraph would be incorrectly hyphenated in the rejustification process, necessitating another correction and this could continue through the end of the paragraph.

(16) It would be desirable to move blocks of text or copy around the screen, again by some type of joy-stick operation. Movement should be up, down, right, left, and diagonal. This feature would be especially helpful for the composition of advertising copy.

These features, of course, range from the "absolute-must" to the "nice-to-have" categories. United Press International is well aware that the inclusion of all of these capabilities in a single stand-alone video display terminal could prove economically unfeasible. As the development of the I. S. and R. system continues, and costs are pinpointed, the terminal requirements may be redefined. But for now, we are looking to CRT manufacturers to design and build a display device which will provide both the newspaper editor and the wire service editor with the ability to do his job in a most efficient, and reasonably economical, manner.

Recent surveys on the accuracy of published news stories report the following: (1) more than half of all published stories contain inaccuracies; (2) two-thirds of those inaccuracies are of an "objective" nature (spelling, typographical errors, etc.) The wire service is certainly susceptible to this problem, and we expect that implementation of I. S. and R., with the video display a key item, will provide us with a solution.

In evaluating CRT's for future use, we turned first to our supplier of computer equipment -- RCA. As part of their Spectra 70 product line, they have developed an excellent video display series. Although we are presently evaluating numerous types of display devices because of our unique requirements, we would like to present here the RCA approach to the design of a commercial video display device.

CRT Expertise

The RCA Corporation is a pioneer in the area of CRT displays. Experience in the development of commercial television, military displays, and computer systems has helped to produce the Spectra 70/750 series of video display products.

Much of the development work has been initiated at the RCA Laboratories in Princeton, New Jersey. In addition to the research effort in the area of display devices, the Laboratories are engaged in computer memory work (such as holographic memories) and in computer software development (in a recently dedicated software laboratory). Display projects include: (1) graphics and color terminals; (2) liquid crystal displays, which have produced a high-resolution picture described as the electronic equivalent of the printed page; (3) a cathode ray storage tube that could simplify the display of stock quotations or computer data; (4) the Model-T character graphic display, which accepts coded digital data from a voice-grade line and displays graphical and alphanumeric information on a direct-view storage tube; (5) light-pen operation for time-shared computers and remote, telephone-linked displays. Out of this environment has come the RCA Spectra 70/750 Video Data System.

Two alternatives are available to the computer user who wishes a display capability: A modular video system and a stand-alone display. The former, the 70/751 is designed for the user with a requirement for multiple units in a single location. A controller houses video data generators, which in turn provide delay line memory and control electronics for the attached video data terminals. Both local and remote operations are possible, and a wide range of screen arrays are available. Up to 48 terminals can be attached to a single controller. The latter alternative, a stand-alone unit is completely self-contained. The original entry in this category, the 70/752, has recently been succeeded by a newer unit, the 70/7522, whose features and enhancements provide many of the capabilities which UPI requires. It is this unit which is examined now in detail.

The Spectra 70/7522 VDT

This unit can easily be described by listing, in tabular format, some of the hardware specifications and then describing, in some detail, some of the features and enhancements. First, the specifications:

Screen size - 12" rectangular
Brilliance - High intensity P4 Phosphor
Character Generation - Monoscope
Controls - Focus, Brightness, Fine Tuning
Character Set - 96 ASCII characters (upper and lower case alphabetics, numerics, and specials)
Standard Screen Arrays - 20 lines x 54 characters (character size 0.14" x 0.10")
14 lines x 81 characters (character size 0.12" x 0.085")
Keyboards - Electronic, high performance. Three models:

1. Upper Case only
2. Upper Case + Right Hand numeric (keypunch) keys
3. Upper and Lower Case

Type of Memory - Delay line
Refresh rate - 60 frames per second
Type of Cursor - Non-destructive
Transmission - Synchronous or asynchronous, selectable at installation time
Transmission rate - 300 to 2400 bps (leased voice)
300 to 2000 bps (message network)

Some of the UPI requirements are met by the basic design specifications of the 70/7522 (screen clarity, standard keyboard, 96 character set, upper and lower case, stand-alone unit). Some of the additional requirements have been met by some of the system's enhancements, outlined below:

(1) Data Insert - A character entered from the keyboard is inserted into a line by moving all characters to the right of the insertion point one position to the right. Text at the end of a line spills over to the next line, where existing characters are also shifted to the right. Whole words are not recognized.
(2) Erase Controls - the entire screen, a line or partial line, or a single character may be erased. The latter function closes up the line to the left.
(3) Cursor Movement - Eight controls are provided: Forward, backspace, next line, home (top left of screen), start of previous line, end of previous line, index up, and index down.
(4) Processor Override - a switch will permit the processor to send a message to the VDT while it is in the write mode. This feature will allow, for example, a news flash alert to be sent to the editor who may already be creating another story.
(5) Data Format - Formats (headings, etc.) can be called up from the processor and displayed in lower intensity characters, which cannot be changed from the keyboard, and which remain on the screen when keyboard entries (variable data) are erased. Only the keyboard entries are transmitted.
(6) Printer Adapter - Hard copy device prints the contents of the screen at up to 120 characters per second. Operation can be controlled by the operator, by the processor, or automatically (for unattended operation).
(7) Screen Address - Messages from the processor can be displayed at any point on the screen.
(8) Double Page - Two 1080-character segments of the delay line memory can be accessed. Two "pages" of text can be transmitted to the terminal, and each one displayed in turn at the terminal. Operator may recall the first page, after having replaced it on the screen with the second.
(9) Screen Array Expansion - In addition to the standard screen arrays mentioned above, two additional arrays are available: 1620 characters (30 x54) or 1782 characters (22x81).
(10) Retransmit - The processor can obtain an automatic retransmission of the last message sent from any 70/7522.

This unit can solve some, but certainly not all, of the special requirements of the news wire service. It is flexible enough to permit multiple units on a single line, and it can be operated on the same multi-point circuit with the modular display system.

SUMMARY

Processing of news information is about to enter a new era. In the forefront of this move is the wire service organization. In an effort to provide error-free copy to its clients, UPI will arm its editorial personnel with a video display capability--a direct input/output avenue to its computers. The video display terminal will be geared to the text preparation and correction needs of the wire service editor. The terminal would meet most of the needs of the newspaper editor as well. The UPI requirements listed above will have to be considered in the light of the economics involved. The closer the terminal comes to satisfying these requirements, the easier the task of substituting the display for the editor's pencil.

The Video Display Terminal is the key to UPI's Information Storage and Retrieval system. If our editors cannot or will not adapt to the terminal the whole project will fail.

It is for this reason that we have planned a demonstration system to resemble the final IS&R system as closely as possible and specifically designed to:

(1) Provide a human factors test of the UPI editors' reaction to performing the editing functions on a Video Display Terminal.
(2) Provide a training facility for the editing staff for the practice use of the VDT in a mode of operation that closely simulates the environment of the actual system. Early use of the terminal by the newsmen will provide the opportunity for them to participate in the operational system development. That participation will take the form of their specific comments concerning the physical and operational requirements of the terminal.
(3) Provide an extensive test and shakedown of the overall IS&R system concept and validity before, during the development, and installation of the full system.
(4) Provide additional insight into the most needed and utilized editing functions that the terminal must accommodate.

ACKNOWLEDGEMENTS

This paper was prepared in cooperation with the RCA Computer Systems Division, Cherry Hill, N.J. and Data Dimensions, Inc. of Stamford, Conn.

BIBLIOGRAPHY

1. Editor and Publisher; Dec. 6, 1969.
2. Editor and Publisher; Nov. 15, 1969.
3. Editor and Publisher; Oct. 11, 1969.
4. Data Processing Magazine; Oct. 1969.
5. Data Systems News; Sept. 1969.
6. Datamation; June 1968.
7. Forbes; Oct. 1, 1969.
8. IEEE Transactions on Computers; May 1968.
9. IEEE Transactions on Computers; August 1968.

PUBLISHING SYSTEMS DEVELOPMENT

R. F. Jefferson, Federal Systems Division, IBM Corporation

INTRODUCTION

It is the intent of this paper to describe trends in information technology which the author feels must be applied to automated publishing systems if such systems are to be capable of meeting present needs and, perhaps more importantly, if they are to be capable of supporting expansions into new markets. The approach taken is first, to summarize the characteristics of automated publishing systems currently proposed and implemented; second, to discuss those elements of future systems which will be the key characteristics in future development of these systems; and third, to review the technologies which must be integrated in order to realize systems with the required characteristics.

To demonstrate that the concepts are not merely speculative, an attempt is made to present specific examples of applications where pertinent techniques are being developed and applied.

CURRENT SYSTEMS

Characteristics

The automated publishing systems currently being proposed and implemented can be characterized as follows:

1. Input of copy in a batch mode, utilizing paper tape with occasional online copy entry; numerous commands are entered with the text
2. Hyphenation and justification programs; subsequent proofing from line printer output
3. Batch entry of formatting and layout parameters either in the text stream or as system constants; some online entry of formatting and layout parameters
4. Cold type; photoprinter output of text usually in galley form with subsequent paste-up

The degree of interaction available to the user is minimal or nonexistent. Changes or modifications to text or composition require reentry of some input and subsequent reprocessing to paper tape or film output. Turn-around times are in terms of minutes or, in some cases, hours.

System Description

The algorithmic, text handling orientation of current systems seems to be in logical contradiction with the graphic nature and graphic arts quality of the final product. Furthermore, the batch-oriented approach to processing seems totally at odds with the highly interactive and judgemental business of publishing. These systems are attempting to support a major segment of the graphics business. However, critical system elements such as input and output devices and the application program packages have traditionally been concerned with text only; the use of known computer graphics techniques has been negligible.

Special keyboards and character readers for text entry, photoprinters for text output (in columnar or galley form), and application program packages with little or no interactive capability, are all too typical. CRT display terminals are infrequently used at present. In cases where they are used, it is usually for text only and seldom do the terminals possess a graphics capability. Pagination, layout, and composition are all planned and accomplished by using programmed algorithms. There are no means of visually checking intermediate results and there is no way to perform modifications until the final output is created.

It is the author's contention that current systems are in that state not as a result of the state-of-the-art, but because consideration has not been given to the total job on a systems basis. An attempt is made to illustrate that technology is not the limiting factor later in this paper.

Another major characteristic of current systems is a lack of emphasis on the value inherent in the data we process. While it is still true that "time is money", it is even truer that "information is money." At conferences dealing with publishing, a considerable amount of discussion is generated concerning the role of the publisher in the information business or the communications business. We talk about publishing systems of the future in which the various print media and distribution techniques will merge in a technical sense as well as a business sense. We envision total systems (device independent and media independent) which will generate new and specialized information products. Are current systems being planned for this? With few exceptions I would say not.

One related specific that should be addressed here is the matter of resource management. Any publishing system, automated or manual, is faced with a massive resource management problem. Even if one ignores the whole area of physical resources used for production and distribution (i.e., presses, paper, and trucks) there is still the major problem of maintaining control over that transitory and valuable commodity, information. The volume, variety, and complexity of the input and output in a publishing system represents a technical and business problem which ranks with the best. Add to that the time constraints and economic ground rules associated with that information and the problem becomes even more complex. The flow of news, in-house and by wire service, the classified and display ads, the cuts of all kinds, the layout information, the production requirements and limitations must all be efficiently controlled and managed. Automation can bring dramatic improvement in this most critical area.

This is an area where the business objectives of publishing can be implemented without interfering with and in fact enhancing the objectives of turning out a good product. Despite all this, automated publishing systems as currently implemented largely ignore this area.

In summary, current publishing systems have been deficient in two areas:

1. System implementation has concentrated on the noninteractive text handling aspects of the problem, relying on programmed algorithms and manual paste-up to produce the graphic products
2. Implementers have not stepped up to the business potential of the information being processed nor have they addressed the problem of resource management

FUTURE SYSTEMS

In discussing future systems, the primary emphasis should be on those characteristics which will make those systems distinct from current systems. The future is defined here as one to five years from current and hence is practical and can be realized in the short term. No technological breakthroughs are presupposed, only thoroughgoing systems development.

System Characteristics

The automated publishing systems which are being developed and which will be implemented during the next five years are characterized as follows:

1. Input of text and data lists continuing in batch mode from paper tape will gradually be supplemented by online keying of time and critical text
2. Direct entry of remotely prepared text through communications facilities and, in some systems, the use of Optical Character Readers
3. CRT displays for text correction and verification
4. Full page online composition through CRT displays; edit and layout in the graphic mode
5. Full page photoprinting – output of a full page image on film or paper as composed in (4) above
6. Inclusion of half-tones and graphics in (4) and (5) above
7. High speed, high resolution, low cost image transmission for satellite printing
8. Remote inquiry, editing, and layout capability through CRT displays
9. Online edit and manipulation of graphics and half-tones including cropping, scaling, and changes to engineering drawings
10. Data bank generation and maintenance
11. Specialized products extracted from the Data bank and "published" in media-independent forms
12. Online validation of data content and source: Data Bank consistency, background retrieval, and credit checks.

System Description

To illustrate the nature of systems which might be developed with the characteristics listed above, several block diagrams may be useful. Figure 1 is basic diagram which in various forms is well known and well worn. While it raises nearly as many questions as it answers, it does show the basic functions which an automated publishing system must perform. The system must handle input; it must provide a facility to edit the data, hyphenate and justify it, and in some way compose it; cold type output is a necessity, so photocomposed output must be provided; finally the film must be converted to a printing plate.

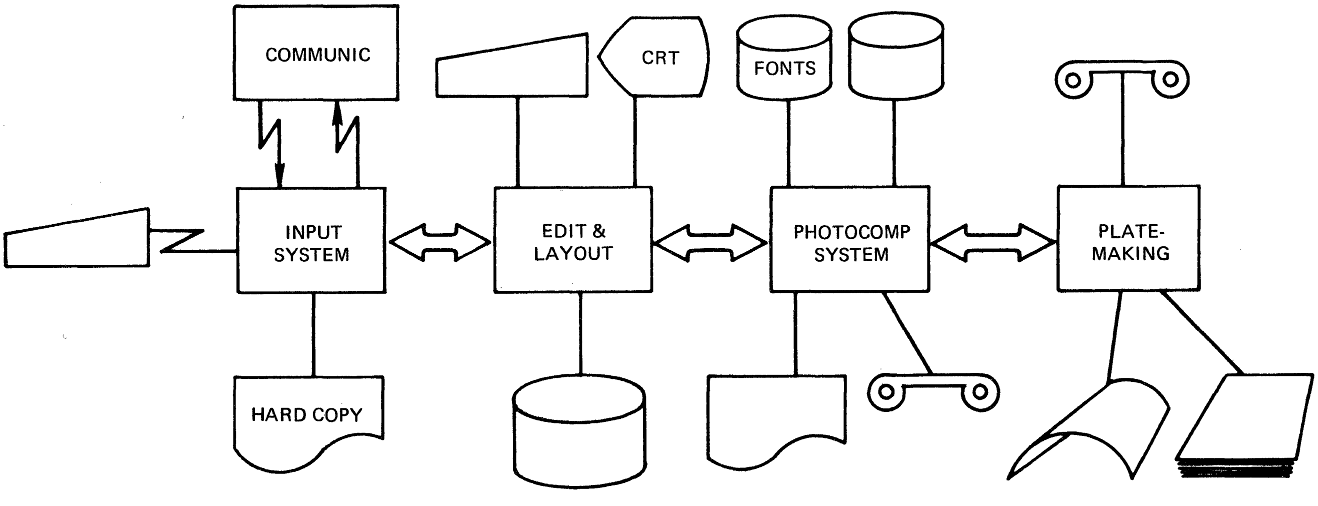


Figure 1. Automated Publishing Systems

These are logical breakouts of the problem and while the diagram in no way represents a system (perhaps four systems), it does allow us to focus on the hardware items necessary to implement a system. These include special keyboards for input and edit, CRT display terminals, optical character readers, scanners, photocomposers, tablets, and digitizers.

From a systems standpoint, however, it is far better to understand the internal operation of such a system before too much time is spent on exotic hardware devices. Figure 2 shows a data flow diagram for a single system with similar objectives to the four systems shown in Figure 1. The same functions are shown, including data capture, correction, layout, composition, output, and platemaking. However, there are definite differences between this and the previous diagram:

a. The data flow within the system is crystal clear. There are only two logical files: the text file and the layout file. All functional elements of the system must be capable of high-speed access of these files.
b. Resource management and production control is exercised by assigning information to a structured data file with the hierarchical relationships stated explicitly and implicitly. The progress of the data can be recorded and controlled as it moves through the system; priorities can be automatically assigned and executed; and an interactive method of modifying priorities can be implemented.

A final difference between the two diagrams is that a logical split is shown in the file between the text and the layout parameters with several access paths to each. This is a key difference which occurs in few current systems. The batch-oriented text processing systems by and large place layout and composition commands within the text, or at most assume that such commands are already loaded and are waiting to descend upon the unsuspecting text. Even where text can be modified in an interactive mode, layout and composition parameters cannot.

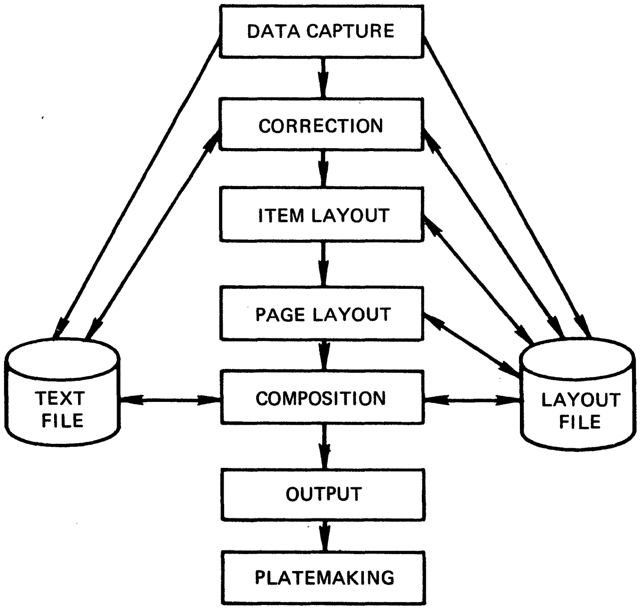


Figure 2. Data Flow Diagram

This follows from the earlier claim that graphic technology has made few inroads into current automated publishing systems. If there is no way to visually review intermediate layout and composition results, there is less need to modify the parameters employed in that process.

Clearly one block diagram cannot describe a system. Neither can it ensure that all of the magical properties implied by the term "total system" will accrue if such a diagram is used. Nevertheless, it does serve to illustrate several of the critical requirements previously discussed which could permit analysis of such a system.

A slightly different view of this system, Figure 3, demonstrates that the data flow follows the same basic pattern as Figure 2. (The system files are shown on the left; the various functions which are to be performed are shown on the right; and the devices used to accomplish the functions are shown in the center.) This block diagram also illustrates the key points in the process where a CRT display terminal would be useful as an interactive graphic device. Text correction by CRT is commonly done and is well understood; item layout and full page layout using a graphic display terminal is not so common. The objective in both cases, is to create a page image which can eventually be committed to print. The graphic display provides a tool by which layout parameters can be entered and intermediate layout results can be reviewed.

The use of a CRT display terminal to create a page image involves additional benefits quite aside from composing a page of printable text more efficiently than current systems. System designers can immediately begin to consider the advantages of including cuts, graphics and half-tones, in the CRT layout process.

To accomplish such a goal, the system must have a way of capturing and holding the cuts; the CRT display must have a way of portraying and manipulating the cuts; and the photocomposition output device must have a means of merging the graphics with the text for output. The data flow diagrams shown in Figures 2 and 3 must be added to as shown in Figure 4 in order to accommodate this next step. Critical to all this, however, is the point that while this is a very significant step in terms of publishing systems capability, and a major effort in terms of system development, it is in no sense outside the scope of existing technologies.

Information Systems

On the related matter of being the information business or the communications business, the point made earlier is that current systems do not provide a mechanism for expanding into new information areas quickly enough. It is possible, however, to point out some instances where progress is being made in this direction and to indicate some related application areas where there is even greater progress.

Well known examples of current publishing applications where auxiliary information is used to support in-house services include the credit checking systems currently being implemented at several large metropolitan dailies. Auxiliary customer files keyed to the classified ad production files as well as to the billing and accounting files are used in a sophisticated way to support the business objectives of the newspaper. Subscription fulfillment, subscription analysis, marketing studies, and mail room automation are still other examples of information control systems in current use.

In the category of peripheral uses of production information, a prime example would be an information services system such as is being implemented at the New York Times. This system will utilize the New York Times index as the key to an historical data base of the items which have appeared in the newspaper. Supporting the application is a unique hardware configuration utilizing video display terminals for access to a computer data base and an image file.

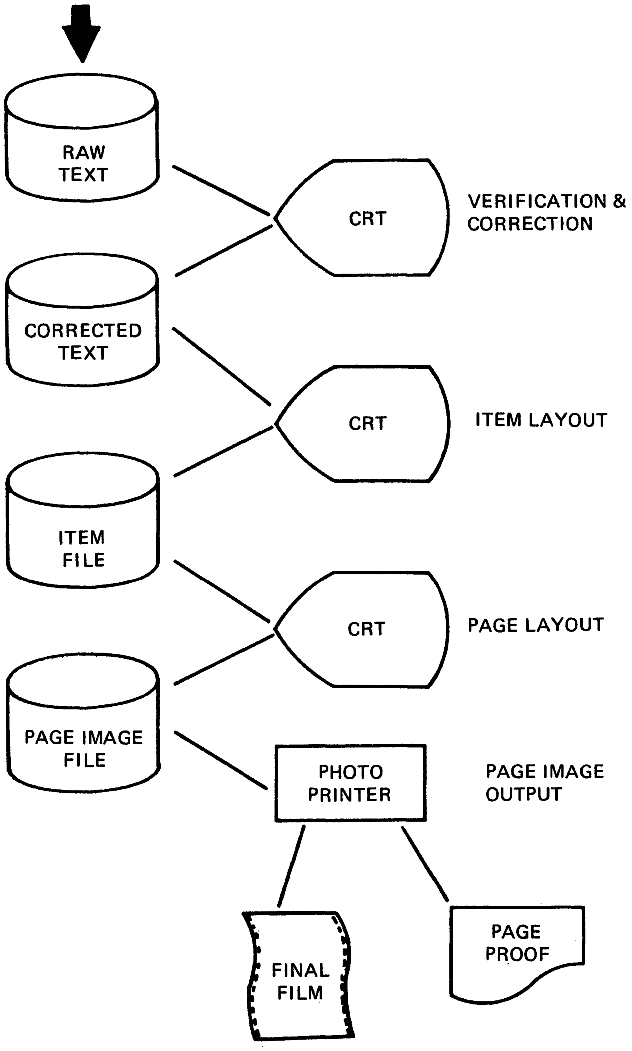

RAW TEXT
CRT
VERIFICATION & CORRECTION
CORRECTED TEXT
CRT
ITEM LAYOUT
ITEM FILE
CRT
PAGE LAYOUT
PAGE IMAGE FILE
PHOTO PRINTER
PAGE IMAGE OUTPUT
FINAL FILM
PAGE PROOF

Figure 3. Edit and Layout Data Flow

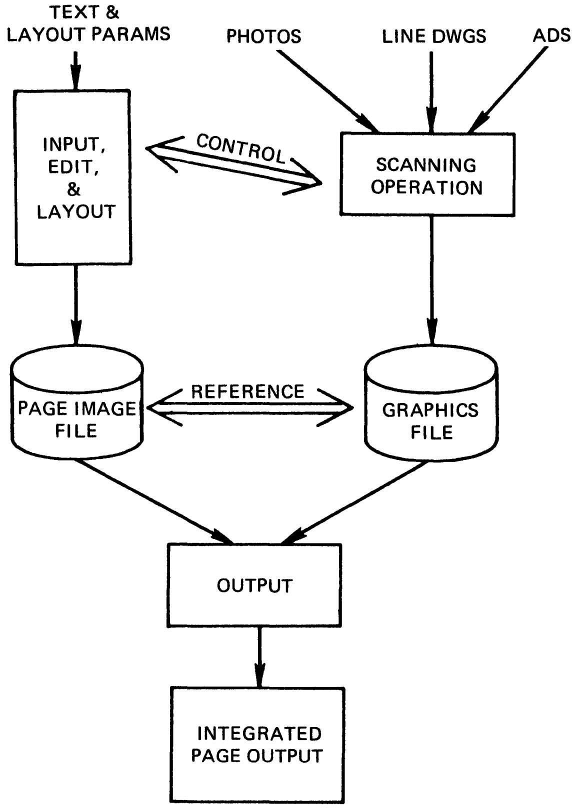

TEXT & LAYOUT PARAMS
PHOTOS
LINE DWGS
ADS
INPUT, EDIT, & LAYOUT
CONTROL
SCANNING OPERATION
PAGE IMAGE FILE
REFERENCE
GRAPHICS FILE
OUTPUT
INTEGRATED PAGE OUTPUT

Figure 4. Integrated Page Composition

Initially, this information services system will be used in-house but it will eventually be marketed to businesses, government agencies, universities and libraries, and research consultants. Terminals installed on customer sites will provide direct access to the Times' data bank. This certainly illustrates the potential which one company feels is in the information services business.

As another example, publicity has been given to a home facsimile system presently under development by a newspaper in Japan. The advent of the home delivery system seems much closer when such technology can be seen and touched.

The key to all of this, of course, is the construction of a data base which is capable of supporting such ambitious plans. Such a data base must deliberately structured to support new technologies and new information products. Methods of extracting, restructuring, and recombining information on a dynamic basis must be planned for now or such systems will not be realized later.

A parallel application is the work currently going on to structure cartographic/land use data banks. The objectives are familiar: structure a data base which is capable of supporting the publication of graphic products, namely maps, and which at the same time can satisfy the immediate and specialized information needs of other users (in this case city planners, developers, highway engineers, and land management people). Systems work is now going on which demonstrates that implementation of such a concept is possible.

A prime example of resource management and information control in an industry related to publishing is the NHK broadcasting system in Japan. This system, called TOPICS, controls and manages a wide variety of resources related to the operation of two television networks and three radio networks. The system maintains information files involving every aspect of broadcasting production, including scheduling, budgeting, inventory control, and allocation of resources. The extensive use of CRT display devices supports the highly volatile and interactive nature of the process—a process which has many parallels in the publishing industry.

NECESSARY TECHNOLOGIES FOR FUTURE SYSTEMS

In order to realize the kinds of systems described above, a number of related but diverse technologies must be brought together. The predisposition of most implementers towards hardware technologies should not be allowed to override a critical fact: if such systems are to be realized, the critical path is software development.

A representative list of technologies necessary for automated publishing systems includes:

1. CRT displays: alphameric and graphic data in digital and video form, configured in both large and small clusters, and capable of local and remote attachment.
2. Large capacity film files: high resolution microform storage, computer controlled access, read station interfaced to CRT display system, local hard copy.
3. Film Scanner/Recorder: scanning of 16, 35, and 70 mm film under computer control, 4000 line scan image, programmed manipulation of scanned input, film output (recording) to same media and resolution.
4. Drum Scanner/Recorder: large area scanning/recording, up to 1000 lines per inch resolution, programmed manipulation of scanned input, recording to same media and resolution.
5. Wideband communications technology to support all of the above
6. Extensive system software development with emphasis on graphics and facsimile processing software.
7. Formatted file systems for information handling functions associated with the data bank.

Implied in all of the above is a central computing facility with the necessary peripherals, an outstanding set of system designers and implementers, and sufficient time and resources to bring everything together into a system which will accomplish the objectives set forth previously.

CONCLUSION

It is not necessary to acknowledge that additional development is necessary in many of the areas discussed, in order to realize complete working systems such as those described. It must, however, be stressed that no technological breakthroughs are necessary; the technologies exist. What remains to be done is to bring these technologies together in an efficient system. The tendency in the past has been to stress areas of hardware development such as keyboards, optical character readers, and photoprinters. Software developments have largely dealt with the text processing aspects of the problem (i.e., hyphenation and justification.) The need is to stress the systems aspects of the total problem from an information handling standpoint and to stress the obvious graphic nature of the information with which the systems must deal.

APPENDIX

In order to more fully explain and illustrate the kinds of systems which are currently utilizing the technologies and approach characterizing "future" systems, it is useful to select three with which the author has first hand familiarity. While none of these address the production aspects of an American newspaper or publishing house, many of the concepts, functions, procedures, and data structures are pertinent. All three systems illustrate the use of interactive displays in a publishing environment.

Coast and Geodetic Automated Cartographic System

Several years ago the IBM Federal Systems Division worked jointly with IBM's Washington Scientific Center and ESSA's Coast and Geodetic Survey to demonstrate the feasibility of automating a cartographic system. A feasibility demonstration was developed to illustrate the revision and updating of aeronautical charts. These charts are produced by the Coast and Geodetic Survey which publishes the 2500 charts that portray the radio navigation and air traffic control systems for the United States. The charts must be periodically revised and updated at intervals ranging from 28 days to one year.

Subsequent to the development of the demonstration for the C&GS, a pilot system was implemented to further evaluate the concepts. This pilot system had the following essential elements:

1. A System/360 Formatted File System (360/FFS) data base which contained information about the location of basic elements such as airports, radio frequencies, airways, etc. System/360 FFS programs were used to build the files and perform the file maintenance, updates and revisions, that must be made to the data base.
2. Extracts of the changes to this data base are provided to the editors during the update process. The editor retrieves the graphic file for the chart and displays the chart area on the IBM 2250 Display Unit. He makes the necessary modifications using the light pen, program function keyboard, and typewriter keyboard.
3. After modification of the chart, an updated graphic file is available. The data from this file is used to drive an optical plotting device to produce new color separation negatives.
4. The color separation negatives are used to pro-

duce lithographic printing plates for the final chart printing operation.

The system automatically performs the simpler changes and requires only editorial review. Changes in a radio frequency, for example, can be accomplished by replacing the old frequency with the new one. When the change is more extensive and involves moving, enlarging or decreasing the size of a chart element, editorial decisions are required. The editor can control the placement of the element, formatting of the text, font style and size, color and other characteristics of the elements in an interactive mode through his display terminal.

Examples of the text editing and manipulation functions that can be performed using the IBM 2250 Display Unit are as follows:

1. Center Text
2. Align text with another element
3. Change font or point size
4. Break text into segments
5. Change color
6. Add or delete text

While these functions are designed for updating an aeronautical chart, it is possible to visualize how they can be extended to other publishing applications such as display ad editing.

The New York Times Information Bank

A slightly different application is represented by an information retrieval system that is being developed by The New York Times. The New York Times is automating their morgue (library) operations where they maintain source material and copies of articles that have appeared in back issues of The New York Times. The Times produces a bi-weekly printed index to these articles which is distributed to the various users that need to obtain information from the data bank. In the manual system, each article is indexed and an abstract is written describing the contents of the article. A reference to the article is contained under each of these index terms.

The automated system is similar to the manual system in its basic principles. The user will enter key words or index terms through an on-line video terminal, Digital TV, to identify the kind of information that he is interested in obtaining from the data bank. An information retrieval program will make a search of the digital data base identifying articles that are relevant to the request. The abstracts of these articles are displayed on the user's terminal so that he can review them and select those that are of interest to him. If he wants to review the full text of the original articles, he can indicate this on his terminal. Software residing in the IBM System/360 computer will then issue a command to a microfiche film file, where a copy of each article is stored on microfilm. The appropriate microfiche is selected from the file, positioned in front of a TV camera, and the video image of the article is transmitted to the user's terminal. The user can view this microfilm in video form on the same terminal that he used to enter his index terms and review the abstracts. The user has a zoom and slew control available at his terminal providing the capability to position the microfiche in front of the TV camera or to change the focal length of the lens zooming in on a particular portion of the original article.

This system utilizes a unique combination of digital technology to search a large index and television technology to display the information contained in a microfiche file. The advantage of this system is that the user is making an on-line search of the digital index and subsequently making an on-line retrieval of the information contained in the microfiche file. His turn-around time is measured in terms of seconds rather than the hours or days in the previous manual system.

LOW-COST SOURCE DATA TERMINAL FOR TECHNICAL TEXT*

Loren P. Meissner, Mansfield L. Clinnick, and Robert A. Belshe
Lawrence Radiation Laboratory, University of California, Berkeley, California 94720

ABSTRACT

A low-cost source-data entry station, for the entry of technical text into computer files, has been assembled from commercially available hardware, specially constructed interfaces, and carefully tailored software. A keyboard is provided for entry of Latin and Greek alphabetic characters in upper and lower case, as well as numbers and about 100 special symbols. As the characters are entered, they are displayed on a CRT screen by a software character generator which can also display any character in superscript, subscript, italic, or bold face. Control keys are provided for specifying insertions, deletions, and other text modifications with direct display on the CRT screen. A facsimile recorder provides reasonably fast low-cost hard-copy output with adequate resolution. This paper describes the configuration of the hardware presently in use, as well as projections to a "commercially feasible" system which would consist of a PDP-8 computer with 8-K core, 32-K disc, one or more terminals (each consisting of a keyboard and a storage scope), a facsimile recorder, and the necessary interfaces. Hardware characteristics of the interfaces are described, as well as the overall system design of the software and the data structures used.

INTRODUCTION

The usefulness of a digital computer in the processing of text has become widely recognized. However, hardly any of the applications so far implemented have included adequate provision for the incorporation of technical text—that is, of text which may include very many different nonalphabetic symbols and characters, and which may include displayed formulas and other constructions not readily handled "one line at a time" in the standard way.

The source of difficulty in processing technical text is not the output device, for photocomposers with the required capability (such as Linotron 505, Videocomp 830, Photon 713, and Fototronic 1200) are now available which can be driven from magnetic tapes written directly by almost any computer. Admittedly, these machines generally operate somewhat inefficiently on complicated text, but even the slower speed of operation that accompanies the increase of flexibility in handling technical text is acceptable and is far superior to any previous alternative.

Bottlenecks due to computer software are not especially difficult to overcome. It is estimated that one-half man-year should suffice for an experienced programmer with sufficient knowledge of graphics requirements to produce a program which will convert strings of technical text from any one specific, well-designed internal computer format to an output form that will meet the requirements of any specific photocomposer model.

Rather, the difficulty has been at the entry point, and seems to be centered in the need to provide adequate "feedback" to allow an operator to enter technical text reasonably rapidly and with a reasonably low error rate on a keyboard with a repertory containing several hundred different characters.

Providing feedback to the operator of a keyboard with a large character repertory seems to require character images generated by software rather than hardware; therefore, the keyboard must be operated on-line. Accordingly, we are not considering the design of an entry terminal for technical text as an isolated problem, but rather as a part of a complete on-line system including a computer, keyboards with feedback displays, a hard-copy ("proof-mode") output facility, and possible intercommunication links for transmitting data to and from other points of application.

BACKGROUND OF THE PROJECT

The Nuclear Chemistry Group at the Lawrence Radiation Laboratory has published, since 1940, six editions of the nuclear data compilation, "Table of Isotopes." Each edition has approximately doubled in size, and the sixth edition,[1] which was published late in 1967, constitutes a volume of nearly 600 large pages of closely spaced tables and diagrams. Some two man-years were spent in transcribing the textual data for this latest edition onto paper in the desired form, using a modified typewriter with a few interchangeable type bars and with bold-face numbers in the upper-case position on the numeric keys. This edition had finally just gone to press when the authors realized that the entire input operation would have to be repeated "from scratch" for the next edition. The magnitude of this input effort, along with the lag time between the cutoff date for the inclusion of new references and the date of

---

*Work done under auspices of the U.S. Atomic Energy Commission.

publication, led them to examine the feasibility of keeping the data on a continuously updated computer file. The specifications for this file,[2] as they began to be developed, drew major attention for the first time at this Laboratory to the need for considerable improvement in our ability to enter technical text into computer-readable files.

Other potential applications at this Laboratory, for any technical text entry terminal which might be developed, were immediately evident. These included, in particular, a somewhat similar Nuclear Physics data compilation, "Review of Particle Properties"[3] (for which the file would be smaller in volume, but higher in acitvity), as well as technical papers being prepared for publication, source-language ALGOL, and various kinds of bibliographic and other information which was already being processed by computers for the Technical Information Group. Furthermore, it was foreseen that any results that might be obtained here could be of value in the design of similar systems at other institutions.

Our requirements for technical text input capability, then, have included as a minimum the ability to enter the text strings of the "Table of Isotopes" into a computer file. Some of this text is shown as an example in Fig. 1. It is because of this orientation that we have at first rather neglected some other interesting format problems: for example, we have not yet taught our system how to handle complicated mathematical formulas such as displayed fractions.

In 1968, we designed and constructed a keyboard (see Fig. 2) by means of which we could generate text like that in the "Table of Isotopes." We used this keyboard to explore some ideas about keyboard design and arrangement, as well as to generate some text for testing output techniques. The keys of this keyboard are connected to the columns of a diode board, the rows of which determine the output code. The keyboard has 73 "character" keys, with expansion room for about 30 more (see Fig. 3). Upper and lower case Greek characters are superimposed upon 24 of the alphabetic keys. This first keyboard is interfaced to an IBM-526 keypunch in such a way that two card colums are punched each time a key is pressed, and the printing unit of the keypunch is used to provide some minimal feedback to the operator. We have found that this feedback is adequate for rather simple text.

We are actually using this keyboard for entering bibliographic data, some of which has been included in reports published by the Laboratory since the middle of 1969. These publications are produced by reading the cards from the IBM-526 into a large computer, which generates magnetic tapes for direct input to a Linofilm Photocomposer. We have also generated some samples of more complicated text in order to test output techniques. Among these samples we included the portion of the Table of Isotopes shown in Fig. 1, which we converted to magnetic tape form for the Linofilm Photocomposer (Fig. 4) and directly to graphic form via the CRT Microfilm Plotter and the CalComp Incremental Plotter (Figs. 5 and 6). For this complicated text, however, we found that many iterations through the entire system (including CRT output, for example) were necessary for finding and correcting errors which occurred at the keyboard but escaped detection at that point because of inadequate feedback.

About this time, a small computer (PDP-8) with a disc and a storage scope became available for our use on this project, so we constructed a second keyboard (see Fig. 7) to be connected to that computer with the idea of using the storage scope for immediate feedback.

The storage scope has proved adequate for display of strings of text in order to provide immediate feedback (see Fig. 8), but it does not have enough resolution for "proof-mode" page composition, which requires a reasonably accurate representation of the appearance of an entire page, including relative character widths, justification of lines, paragraph indentations, columnar formats, and the like. Because of the complexity of the characters used (see Fig. 9) and the need to distinguish similar characters such as y versus γ, v versus ν, and boldface versus lightface, it is necessary to use an average character width at least 12 — 16 times the minimum resolution of the output device. Thus, in order to simulate an "average" 100-character line, for example (corresponding to a 38-pica line of nine-point type), 1200 — 1600 resolvable points across the screen are needed and a 1024-by-1024 CRT is inadequate.

We wondered whether some hard-copy output device could give us a solution to this problem while also satisfying other possible needs for copy to be proofread off-line. For example, the scientist responsible for the accuracy of the data in the Table of Isotopes needs to be able to review it after a typist has performed the initial data entry. We rejected, on the grounds that they are too expensive, such devices as high-resolution CRT systems (for example, Stromberg Datagraphix 4060, $275,000), low-cost photocomposers with magnetic tape input (for example, Linofilm, $65,000), high-speed incremental plotters (for example CalComp 665, $11,275), and multiple-stylus electrostatic recorders (for example, Varian Statos-5, $18,500). Finally we discovered a 19-inch facsimile recorder which is reasonably fast, has a resolution of about 1800 points per scan line (by 100 scan lines per inch), costs less than $2500 (plus interface), and seems to have further applications as a general-purpose graphical output device.

We proceeded to design interfaces and to develop software to combine the PDP-8, a keyboard, a storage scope for feedback, and a fascimile recorder for proof-mode output, into a technical text entry system. All of the hardware was obtained commercially, except the keyboard and the interfaces.

DESCRIPTION OF PRESENT HARDWARE

Because we believe that a number of organizations have somewhat the same needs as our Laboratory, we have been strongly constrained, as a matter of overall project philosophy, to make our explorations within the area of commercially available hardware wherever possible. On the other hand, the small computer which has so far been available for our use was acquired for the benefit of other users as well; therefore, we have not always been able to proceed in exactly the way which our overall philosophy would most strongly indicate. However, actual departures have been relatively minor and of such a nature that it is fairly easy to project from the actual configuration to a "most commercially feasible" configuration. Furthermore, the philosophical disadvantage of being unable to completely specify hardware is heavily overbalanced by the practical advantage of having the major hardware components already available for our use.

The present hardware configuration consists of a PDP-8I Computer with 8-K core and 256-K disc. It is equipped (because of the requirements of the other users) with a "generalized display interface" and a "generalized serial input-output interface." Both of these interfaces were designed and constructed at this Laboratory. The display interface provides connections for a number of Tektronix "Type 611 storage display units" (storage CRT scopes), including digital-to-analog converters, an "Erase" function output, and "Erase Complete" return signal from the scope, and a "Write-Through" mode (display point on screen temporarily, without storing) which is used especially for cursors or pointers to particular positions within the display. The serial input-output interface provides connections for a number of Teletype-like channels of various speeds: by providing the proper clock-pulse generator, any channel can be made to operate at a specified speed up to about 50 kHz. A seven-track magnetic tape drive is also connected to the computer. The only additional hardware required by our project consisted of a keyboard and a facsimile recorder.

The keyboard which is presently connected to this system was constructed here, using reed switches obtained from Microswitch (a division of Honeywell). It is similar in external appearance to the first keyboard (which drove an IBM-526 keypunch). The main keyboard has two sections. On the left are 49 keys arranged in offset rows like a typewriter keyboard, including 36 letters and digits, 12 keys for commonly used symbols (including punctuation), and the space bar. Greek characters are superimposed upon 24 of the alphabetic keys (all except J and V), and upper- and lower-case Greek letters are engraved on those keys in reduced size above the normal (Latin) letters. On the right, there are 39 keys for additional symbols. If separate characters are assigned to upper and lower case, the 39 keys on the right and 12 keys on the left provide more than 100 different symbols.

Across the top of the keyboard is a row containing four control function keys and ten control character keys. The four control function keys specify Greek, superscript, subscript, and Italic functions, all of which are available along with any of the characters. The other ten keys are "characters" in that they actuate the keyboard output in the same way as any of the normal character keys; however, they are assigned to control specifications such as tab, line feed, and editing controls. Shift and shift lock keys are also provided in the usual typewriter positions.

The 98 "character" keys are encoded, by means of a diode board, into seven parallel bits. The shift, Greek, superscript, subscript, and Italic functions correspond to five more parallel bits. These 12 bits, plus a pulse which occurs whenever a "character" key is pressed, provide the output from the keyboard. The 12 bits could be fed directly to the PDP-8 computer; however, because of the availability of the serial input-output interface, we convert the bits to serial form and send them out on two separate "Teletype" channels.

The facsimile recorder is an Alden* Flying-Spot Component Helix Recorder, Model 319 DA, which for many years has been the standard machine in use throughout the country for receiving wire-transmitted weather maps. It records by applying an electrical signal to moist electrosensitive paper. A mechanical scanning system moves the point of application of the signal along each of a series of closely spaced lines across the paper. The scanning device consists of a straight "knife-edge" electrode above the paper which defines the scan line, and a drum with a single-turn helix below the paper which moves the signal across the scan line each time the drum rotates. The number of scan lines per second is determined by the speed of the helix drive motor, and the number of scan lines per inch (longitudinal resolution) is then determined by the speed of the paper drive motor. Our helix drive operating speed is 600 rpm, or ten scan lines per second, and the paper drive speed is 6 inches per minute, or 0.01 inch per scan (per 0.1 second). A return signal is sent from the rotating drum every time the helix starts a new scan.

The darkness of the recorded image on the paper corresponds directly to the signal level as it varies during the scan sweep. For some applications, such as weather map recording, a slowly varying dc analog signal produces varying intensity in the record: for these applications, a linear amplifier and low-contrast recording paper are used. For digital recording, on the other hand, a precise distinction between black and white is desired, so we use high-contrast paper and a high-gain amplifier, and the applied signal is a pulse train. The frequency of this signal determines the transverse resolution (number of points per scan line). The scan line is 18 inches long, so we use a signal frequency of 18 kHz, or 1800 pulses in 0.1 second, giving 1800 points across the scan line or 0.01 per inch. The longitudinal and transverse resolutions are thus each 100 per inch, which equals the nominal working limit set by the electrical and mechanical characters of the electrodes and of the recording paper.

---

*Alden Electronic and Impulse Recording Equipment Company, Inc., Westboro, Mass.

The 18-kHz pulse train requires that a 12-bit word be loaded into a shift-register buffer every 600 microseconds. The PDP-8 has no difficulty in operating at this rate. Determining the bit configuration to be written requires that the image be generated one scan-line at a time. For text output, this scan generation is considerably simplified by preparing the character images for the original character repertory in suitable form.

PROJECTED HARDWARE CONFIGURATION

Because the present project uses a computer that happened to be available, which was already provided with interconnection hardware that simplified the attachment of some of the input and output devices, it is necessary to make some projections in order to estimate the actual cost of assembling a similar system in a configuration that would be ideal from a commercial point of view. The projections listed here are based on experience with engineering and construction of the hardware we are now using, as well as similar systems developed for other applications at this Laboratory. Cost estimates for a "typical" basic system are as follows:

DEC PDP-8/L computer	$9,150	
Additional 4-K core (including B08A unit)	5,500	
32-K disc (including KD8L)	6,500	$21,150
Tektronix Type 611 storage display unit	$2,500	
Interface (see description below)	2,400	$ 4,900
Microswitch solid-state keyboard (Mono Mode, Type 2-1)	$ 300	
Additional control keys	200	
Interface (see description below)	400	$ 900
Alden Model 319-DA facsimile recorder (with helix drive, paper drive, marking amplifier, magnetic synch trigger, continuous blade drive, and accessories)	$2,395	
Interface (see description below)	1,355	$ 3,750
"Typical" total minimum system		$30,800
Each additional terminal (keyboard and storage scope, with interfaces)		$ 5,800

These costs are intended to apply to acquisition of the components at manufacturer's list prices, and include interface assembly labor. The system design allows for only a limited number of additional terminals, since core and disc space and computer time must be shared by all the keyboards. Four terminals would seem to be a reasonable upper limit, beyond which additional computer capacity would be required.

Interface Descriptions

The interface for the Tektronix Type 611 Storage Display Unit consists of two 10-bit registers (for X and Y coordinates), with digital-to-analog conversion. Signals are provided for Erase, View, and Write-Through (display without storing) functions under control of the computer, and for the detection by the computer of the "Erase Complete" return signal from the display unit. Logic-board costs include $700 for digital-to-analog converter and $600 for display control functions.

The Keyboard interface provides 12-bit parallel input into the Accumulator, with interrupt when a key is pressed on the keyboard. Basic cost for the components of this interface is $225.

The Facsimile recorder interface includes a three-cycle Data Break, Internal Variable Shift Register, and Output Register, at an estimated total component cost of $755. The Data Break is used to control the transfer of words from core memory to the shift register and thus present them to the recorder one bit at a time. Interrupt occurs on "Transfer Complete" or on a Synch Trigger pulse from the Recorder helix drum.

Other costs to be added to these component costs include: (1) a pro-rata share of additional hardware costs including Mounting Drawer Bin, Power Supply, Cables, Connectors, and 10% Contingency (total $1050), and (2) pro-rated assembly labor, design, and drafting charges (total $825). These pro-rated additional costs are included in the cost estimates for the "typical" total system.

Alternatives

It is obvious that the DEC PDP-8 computers are not the only ones to which keyboards (etc.) can be attached. Since the speed of the PDP-8L is not being exhausted, it would be interesting to perform some experiments with a slower computer. As another alternative, the additional 4-K core might be omitted in the case of a one-keyboard system which could avoid overlap of the basic operations (page display, line display with edit, and proof-mode output) each of which uses considerable in-core program space. Additional disc space (at the relatively low cost of $3000 per additional 32 K, up to 128-K total) would be very desirable on multiple-keyboard systems, or for applications requiring direct access to larger amounts of text.

For multiple-terminal systems, a TV-type CRT screen (drive from parallel-head disc tracks) needs to be considered as an alternative to the storage scope. Here the software character generator (inside the computer) would write and update an image of the entire screen on the disc. Cost of the disc storage is about $5000, plus $300 per track. With standard TV resolution (about 200,000 bits per frame), only two tracks per CRT scope would be needed. For resolution comparable to the Tektronix storage scope, on the other hand, about 10 tracks would be required, and other costs would increase because of the need for a higher-quality TV image and because of the higher bit rate (one million bits per frame).

The Microswitch Mono-mode keyboard provides 69 character keys, which would be enough for many applications. Keyboards are now being designed and packaged to order by many other manufacturers, and some of these can provide larger numbers of keys at a reasonable cost.

Unless the proof-mode output is to be used as the final product of this system, there is need for output (and perhaps also input) data channels for communication with various computers or other devices. It is interesting to note, in this connection, that the facsimile recorder interface contains most of the hardware required for an output data channel, and could probably be adapted rather inexpensively to allow the transmission of data from the PDP-8 to other computers. Availability of a magnetic tape drive, on the other hand, would provide considerably enhanced input-output capability for intercommunication.

USING THE TERMINAL

As we have seen, the editing of strings of text from a large and variable character repertory implies software-generated characters. Interactive manipulation of words or lines of text requires on-line display of the completely formed characters, which in turn requires either the storage or the rapid regeneration of the entire picture on the screen. For a small number of terminals, the most economical alternative seems to be a storage scope, with display stored on the CRT screen itself.

When a point has been stored on this kind of scope, it remains there until the scope is erased. Thus, it is easy to add points (or characters, etc.), but it is impossible to delete a portion of the display without erasing all of it. Because of this fact, we were not altogether certain at first that a scheme could be implemented for the interactive manipulation of text by means of this device. The key to success, as it has turned out, is the short time interval required to redisplay a screen full of characters, Thus, it is feasible to delete a particular character, for instance, by erasing the screen and redrawing all of the remaining characters; this requires less than a second. This time interval, however, is still long enough so that we must avoid performing the redisplay unnecessarily, and this fact has had considerable influence upon the methods we used for implementing the interactive manipulation of text with this terminal.

The handling of text at our terminal is analogous to the manipulation of a scroll: the characters that are visible (on the CRT screen) at any given time are in general only a portion of a long string of stored text which may be thought of as being continued to the left and right beyond the field of view. Both the left and right portions of this unseen text are written on disc in 256-character "tracks," with leftover partial tracks (if any) remaining in separate left and right buffers in core.

During the manipulation of text, then, we always divide the entire string of stored characters into a left part and a right part, either of which may be empty. Each of these parts is structured in a simple three-level storage hierarchy. At the highest (most accessible) level we have the few characters which appear on the screen, at the lowest level we have 256-character tracks on disc, and in between are the residual partial tracks which remain in core.

Let us describe the operations the user can perform. We start from the middle, assuming that several hundred characters have been entered and that the text has then been divided (somehow) into a left string and a right string of approximately equal length, so that all the levels of the storage hierarchy are nonempty, including the "left line" of about 20 characters which is displayed near the top of the CRT screen and the "right line" of about the same length near the bottom.

A character can be added to the string by typing on the keyboard in the normal way. The character is added at the end of the left line, and it immediately appears on the screen.

A character can be deleted either from the end of the left line ("delete left") or from the beginning of the right line ("delete right"). In either case, the screen is erased and the display is recreated with the deletion having been performed.

A character can be moved from the end of the left line to the beginning of the right line ("move right"), or vice versa ("move left"). Here again, the CRT screen is erased and a new display is generated.

In all these cases, the text string in storage is modified. The small left or right "line display" storage areas in core can easily overflow or become depleted, and so text is transferred between these areas and the lower levels of the storage hierarchy as required.

Except when characters are being added, the display on the screen reflects the current status of the left and right "line display" storage areas. The regeneration of the display, therefore, restricts the speed of deletions and of moves to about one per second. For this reason, we also have provided a way for the user to delete or move several characters at a time with only one regeneration of the CRT display.

When characters are added, they enter the left line and are eventually stored in the lower levels of the left string. The new characters appear on the screen at the end of the left line, and then on additional lines down the middle of the screen, but the display is not erased until the screen is full.

Initial entry of text is handled by the program in the regular way, except that the right line and the right string at all levels remain empty. At the beginning, the left string is empty too, but as text comes in the left line overflows into the intermediate "partial track" area, and as full tracks are generated they are moved out to the disc. As soon as a "move right" operation occurs, the right string becomes nonempty. A sufficiently large number of these operations will empty the left string and place all of the text in the right string, so that any new data entered from the keyboard will be placed at the very beginning of the text.

In order to speed up the process of locating the desired point within a large body of text, a "page display" mode is provided in addition to the "line display" mode just described. In this mode, no editing is allowed, and the right and left line display storage areas are emptied and pushed back into the track buffers. These are then rearranged so that exactly two full tracks are contained in core, with the rest of the text on disc. These two tracks (512 characters total) are displayed on the screen, with a movable cursor between them. (The cursor is a small mark which is sent to the CRT in "write-through" mode so that it is not stored and must be regenerated several times a second.) The user can move left or right one page at a time: each such operation causes one of the old pages to be written from core to disc and a new page to be brought in, followed by a display of the two tracks currently in core. He can also move the cursor about, within the two-page display. When he returns to "line display" (edit) mode, the text immediately to the left of the cursor is brought into the left line display area and the text to the right goes to the right line area, and the display is recreated as before.

All of the displays described so far use fixed formats, with fixed-width characters and a fixed number of lines on the screen. Line feeds, tabs, etc., do not have any effect on the display except to insert special indicators which appear also as fixed-width characters. Words are broken arbitrarily at the end of the line as well. In short, these displays are strictly "documentary" in function, so that the user can tell exactly what characters are in the string. Even the space bar prints a special indicative symbol.

To better exhibit the ultimate appearance of the text, a "view mode" display is also provided, wherein the characters have variable widths, spaces do not print, and a new line begins at a line feed or between words.

APPLICATIONS

One the text has been stored on the disc, it can be written line-by-line on the facsimile recorder at a rate of about 80 characters per second, or the entire file of 32,000 characters in about 6.5 minutes. After proofreading it at his desk, the user may wish to return to the console and make further corrections. Finally the text (in the form of a 12-bit code for each character) is ready to be used in some way.

For some purposes, the hard-copy output via the facsimile recorder will be the final output from the system. Although the characters generated on the facsimile recorder in the standard "proof mode" are not of the highest graphic arts quality, they are certainly better than the usual (all upper-case, fixed character width, unjustified) computer line printer, and are perhaps somewhat better than typed copy. However, in most cases this hard copy is not the final output; rather, the text entered into the computer is intended for further use in some automatic system, for information retrieval, automatic indexing, language translation, or some other purpose. A somewhat different case is the entry of source-language ALGOL, where the further processing to be performed by the computer consists of compilation and execution of the specified algorithm.

Often the further processing is beyond the capability of the small computer, either because of large storage and file requirements or because of the extent and intricacy of the programs involved. If a seven-track or nine-track tape drive is provided, the text can be transferred from disc to tape and thence to another computer. As an alternative, if the output interface to the facsimile recorder is adapted to provide a data output channel, then the text can be transferred to any computer system with a compatible input channel.

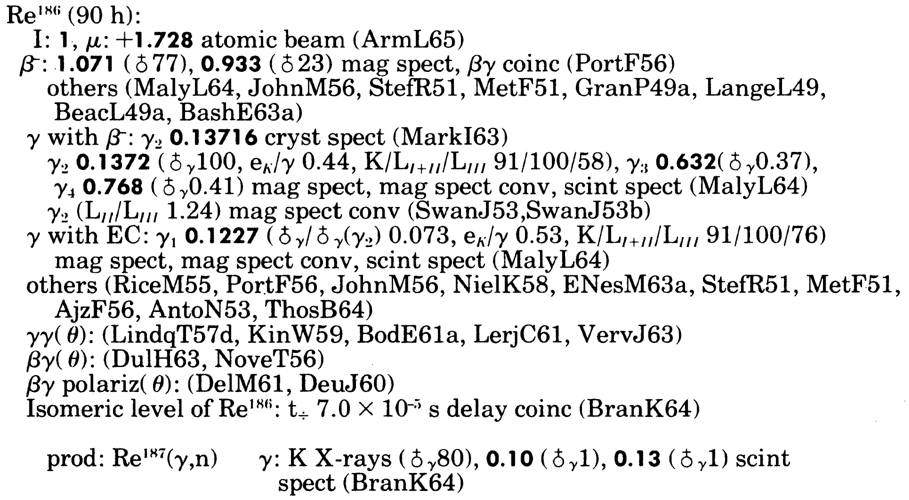
Furthermore, with input from tape or some other reasonably high-speed channel, the facsimile recorder can also be used as a general-purpose graphical output device which can plot some kinds of data faster than a digital incremental plotter. In particular, we use our recorder to plot larger, higher-quality computer-drawn letters and symbols. A page 18 inches wide and 27 inches long, corresponding to a 6×9-inch page after photoreduction, is formatted on a large computer using characters from the repertory designed by A. V. Hershey.[4] Using that storage capacity of the large computer, we convert the data to scan-line format and transmit it in that format to the PDP-8, which sends it directly to the facsimile recorder. The facsimile plotting time for 5000 characters (67 lines of 75 characters each) is 270 seconds or 4.5 minutes, as compared with 20 minutes for the CalComp 665, or about 10 — 12 minutes for the Linofilm Photocomposer. Thus the facsimile recorder, with input from external sources (via magnetic tape and small computer, or otherwise), can be used as a reasonably high-quality photocomposer. This use alone might justify its acquisition by an institution whose report printing requirements amount to a few hundred original pages per week.

We expect to use intercomputer links for typesetting compound fractions and other mathematical material which may be too complicated to analyze on the small computer. We will enter the data in the form of linear strings of text with function codes and delimiters to specify the mathematical nature of the material. Our small computer text entry system will handle it only in this linear string form, and will generate output for analysis by a large computer. The large computer may produce photocomposer output, but for proofreading of the results of the text analysis, we will probably want to simulate the photocomposer by generating the characters and converting them to scan-line format to be plotted on the facsimile recorder.

Another interesting application is the capitalization of existing files of one-case text, such as material which has been keypunched and which contains no upper- and lower-case indications. Such text is processed by a fairly simple program that converts all of the characters to 12-bit form, flagging them as lower case except for certain instances such as the first nonblank character following a period, which makes the correct decision on more than 90 percent of the words in the text. The resulting text is scanned by a human operator in order to correct the errors and oversights of the automatic conversion. This procedure fills a need which exists today, but its importance may be expected to decline as equipment for the entry of text in upper- and lower-case comes into general use over the next few years.

REFERENCES

1.
Lederer, C. M., Hollander, J. M., and Perlman, I., Table of Isotopes, 6th ed. (John Wiley & Sons, Inc., New York, 1967).
2.
Lederer, C. M., Hollander, J. M., and Meissner, L. P., A Computer-Based system for the Nuclear Data Compilation, "Table of Isotopes," Lawrence Radiation Laboratory Report UCRL-18530, October 1968 (unpublished).
3.
Particle Data Group (Rosenfeld et al.): Review of Particle Properties, Lawrence Radiation Laboratory Report UCRL-8030, Revised, 1969.
4.
Hershey, A. V., Calligraphy for Computers, AD-662-398, U.S. Naval Weapons Laboratory, Dahlgren, Va.



Re^{186} (90 h):
I: **1**, μ: +**1.728** atomic beam (ArmL65)
β^-: **1.071** (♁77), **0.933** (♁23) mag spect, $\beta\gamma$ coinc (PortF56)
others (MalyL64, JohnM56, StefR51, MetF51, GranP49a, LangeL49, BeacL49a, BashE63a)
γ with β^-: γ_2 **0.13716** cryst spect (MarkI63)
γ_2 **0.1372** (♁$_\gamma$100, e_K/γ 0.44, $K/L_{I+II}/L_{III}$ 91/100/58), γ_3 **0.632**(♁$_\gamma$0.37), γ_4 **0.768** (♁$_\gamma$0.41) mag spect, mag spect conv, scint spect (MalyL64)
γ_2 (L_{II}/L_{III} 1.24) mag spect conv (SwanJ53,SwanJ53b)
γ with EC: γ_1 **0.1227** (♁$_\gamma$/♁$_\gamma$(γ_2) 0.073, e_K/γ 0.53, $K/L_{I+II}/L_{III}$ 91/100/76) mag spect, mag spect conv, scint spect (MalyL64)
others (RiceM55, PortF56, JohnM56, NielK58, ENesM63a, StefR51, MetF51, AjzF56, AntoN53, ThosB64)
$\gamma\gamma(\theta)$: (LindqT57d, KinW59, BodE61a, LerjC61, VervJ63)
$\beta\gamma(\theta)$: (DulH63, NoveT56)
$\beta\gamma$ polariz(θ): (DelM61, DeuJ60)
Isomeric level of Re^{186}: $t_{\frac{1}{2}}$ 7.0 × 10^{-5} s delay coinc (BranK64)

prod: $Re^{187}(\gamma,n)$ γ: K X-rays (♁$_\gamma$80), **0.10** (♁$_\gamma$1), **0.13** (♁$_\gamma$1) scint spect (BranK64)

XBL 703-462

Fig. 1. Portion of a page from Table of Isotopes, 6th ed. (John Wiley and Sons, Inc., New York, 1967).

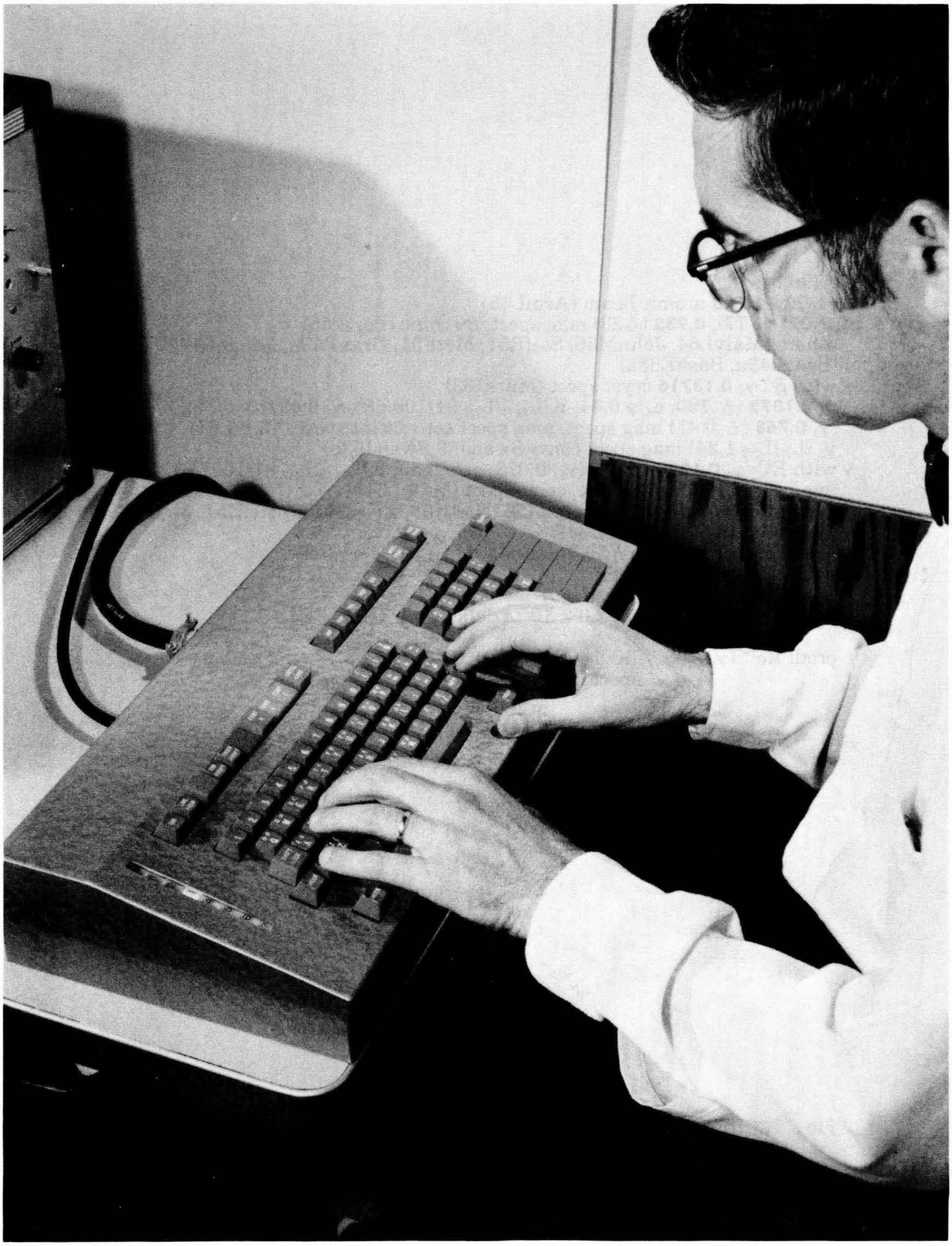
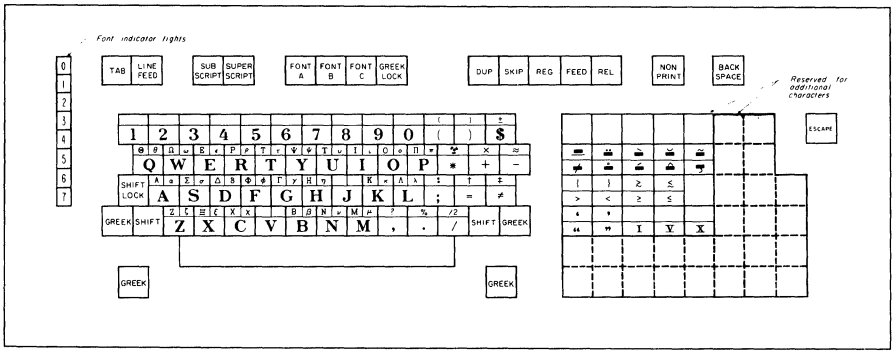
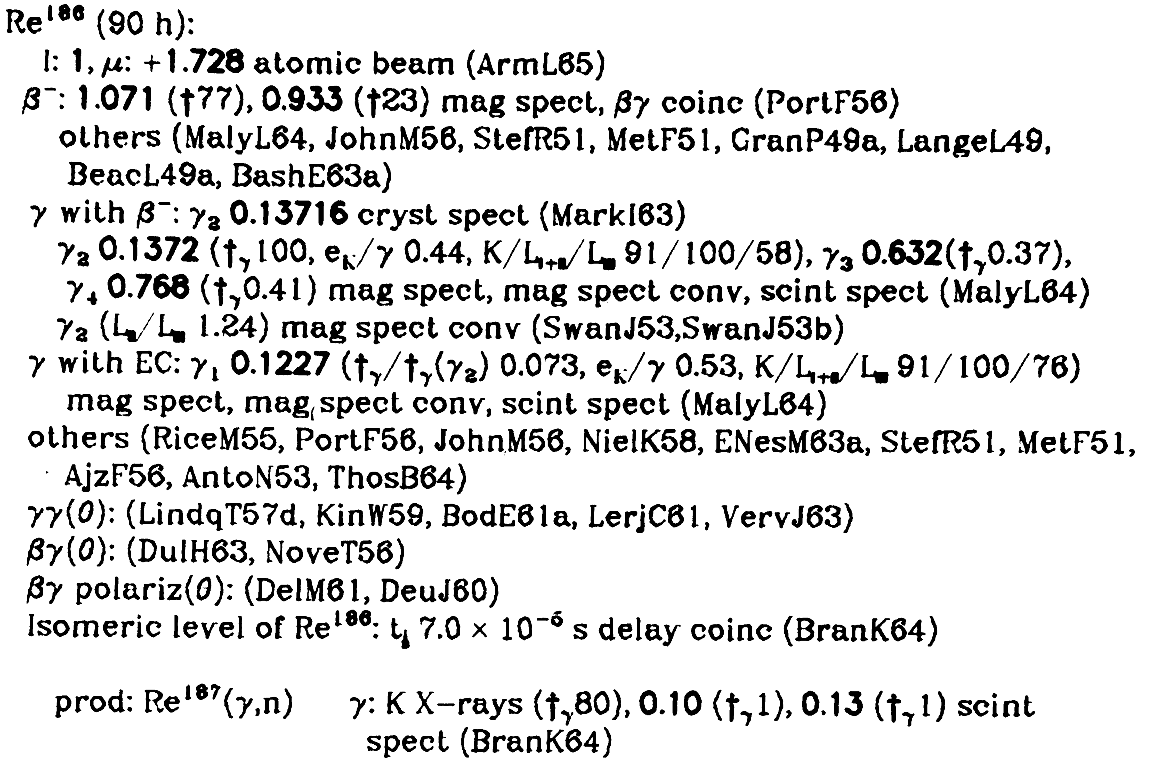


Fig. 2. Keyboard constructed in 1968, which drives an IBM-526 keypunch via a relay-box interface.


Font indicator lights
0
1
2
3
4
5
6
7
TAB
LINE FEED
SUB SCRIPT
SUPER SCRIPT
FONT A
FONT B
FONT C
GREEK LOCK
DUP
SKIP
REG
FEED
REL
NON PRINT
BACK SPACE
Reserved for additional characters
ESCAPE
1 2 3 4 5 6 7 8 9 0 ( ) $
Q W E R T Y U I O P * + −
SHIFT LOCK
A S D F G H J K L ; = ≠
GREEK
SHIFT
Z X C V B N M , . /
SHIFT
GREEK
GREEK
GREEK

XBL6810-7092

Fig. 3. Layout of 1968 keyboard.



Re^{186} (90 h):

I: **1**, μ: **+1.728** atomic beam (ArmL65)

β^-: **1.071** ($\dagger$77), **0.933** ($\dagger$23) mag spect, $\beta\gamma$ coinc (PortF56)

others (MalyL64, JohnM56, StefR51, MetF51, GranP49a, LangeL49, BeacL49a, BashE63a)

γ with β^-: γ_2 **0.13716** cryst spect (MarkI63)

γ_2 **0.1372** ($\dagger_\gamma$100, e_K/γ 0.44, $K/L_{I+II}/L_{III}$ 91/100/58), γ_3 **0.632**($\dagger_\gamma$0.37), γ_4 **0.768** ($\dagger_\gamma$0.41) mag spect, mag spect conv, scint spect (MalyL64)

γ_2 (L_{II}/L_{III} 1.24) mag spect conv (SwanJ53,SwanJ53b)

γ with EC: γ_1 **0.1227** ($\dagger_\gamma/\dagger_\gamma(\gamma_2)$ 0.073, e_K/γ 0.53, $K/L_{I+II}/L_{III}$ 91/100/76) mag spect, mag spect conv, scint spect (MalyL64)

others (RiceM55, PortF56, JohnM56, NielK58, ENesM63a, StefR51, MetF51, AjzF56, AntoN53, ThosB64)

$\gamma\gamma(\theta)$: (LindqT57d, KinW59, BodE61a, LerjC61, VervJ63)

$\beta\gamma(\theta)$: (DulH63, NoveT56)

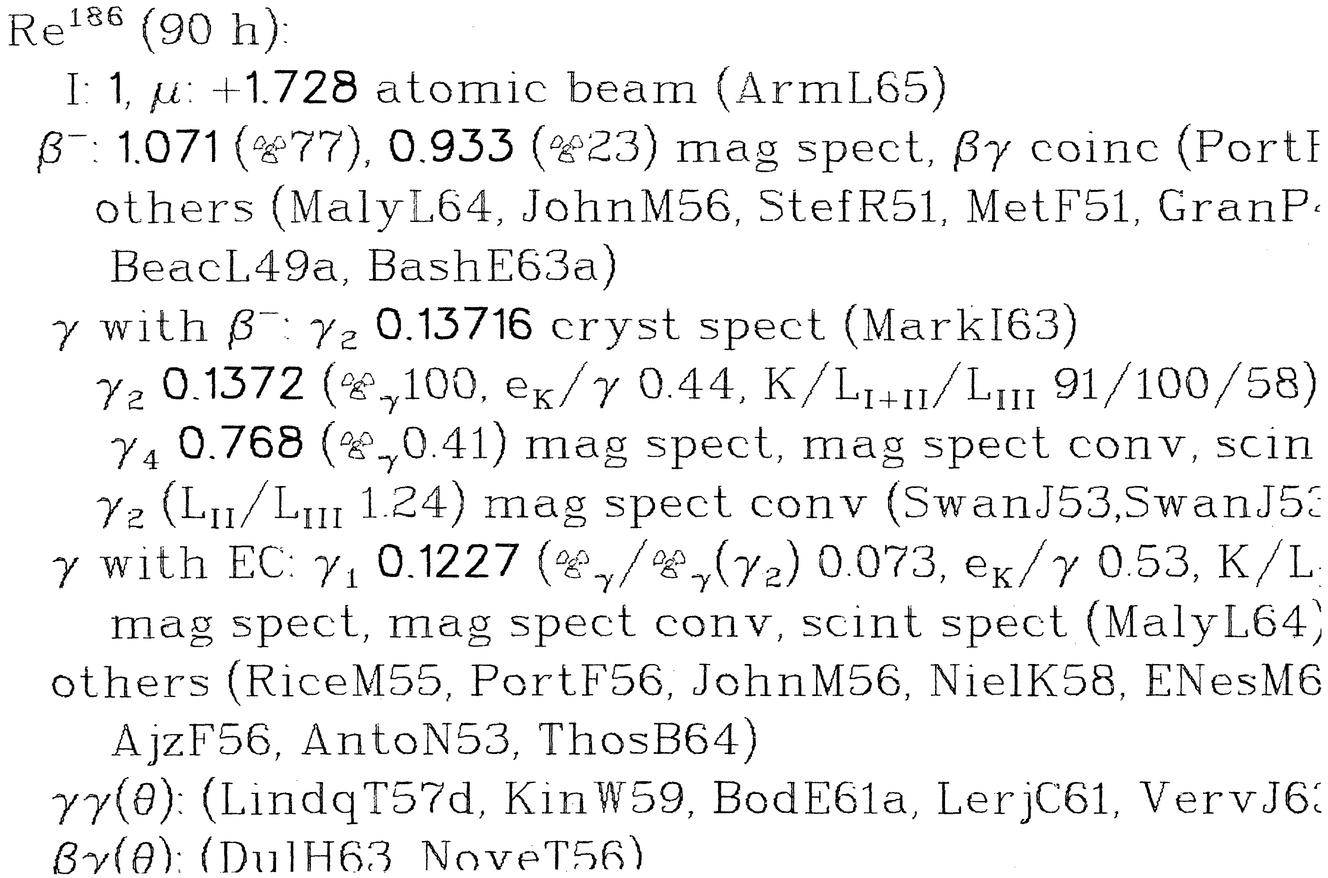
$\beta\gamma$ polariz(θ): (DelM61, DeuJ60)

Isomeric level of Re^{186}: $t_{\frac{1}{2}}$ 7.0 × 10^{-5} s delay coinc (BranK64)

prod: Re$^{187}(\gamma,n)$ γ: K X-rays ($\dagger_\gamma$80), **0.10** ($\dagger_\gamma$1), **0.13** ($\dagger_\gamma$1) scint spect (BranK64)

XBL 703-463

Fig. 4. Sample of technical text photocomposed on Linofilm, directly from computer-generated magnetic tape.



Re^{186} (90 h):

I: 1, μ: +1.728 atomic beam (ArmL65)

β^-: 1.071 ($\mathscr{I}$77), 0.933 ($\mathscr{I}$23) mag spect, $\beta\gamma$ coinc (PortF

others (MalyL64, JohnM56, StefR51, MetF51, GranP

BeacL49a, BashE63a)

γ with β^-: γ_2 0.13716 cryst spect (MarkI63)

γ_2 0.1372 ($\mathscr{I}_\gamma$100, e_K/γ 0.44, $K/L_{I+II}/L_{III}$ 91/100/58)

γ_4 0.768 ($\mathscr{I}_\gamma$0.41) mag spect, mag spect conv, scin

γ_2 (L_{II}/L_{III} 1.24) mag spect conv (SwanJ53,SwanJ5

γ with EC: γ_1 0.1227 ($\mathscr{I}_\gamma/\mathscr{I}_\gamma(\gamma_2)$ 0.073, e_K/γ 0.53, K/L

mag spect, mag spect conv, scint spect (MalyL64)

others (RiceM55, PortF56, JohnM56, NielK58, ENesM6

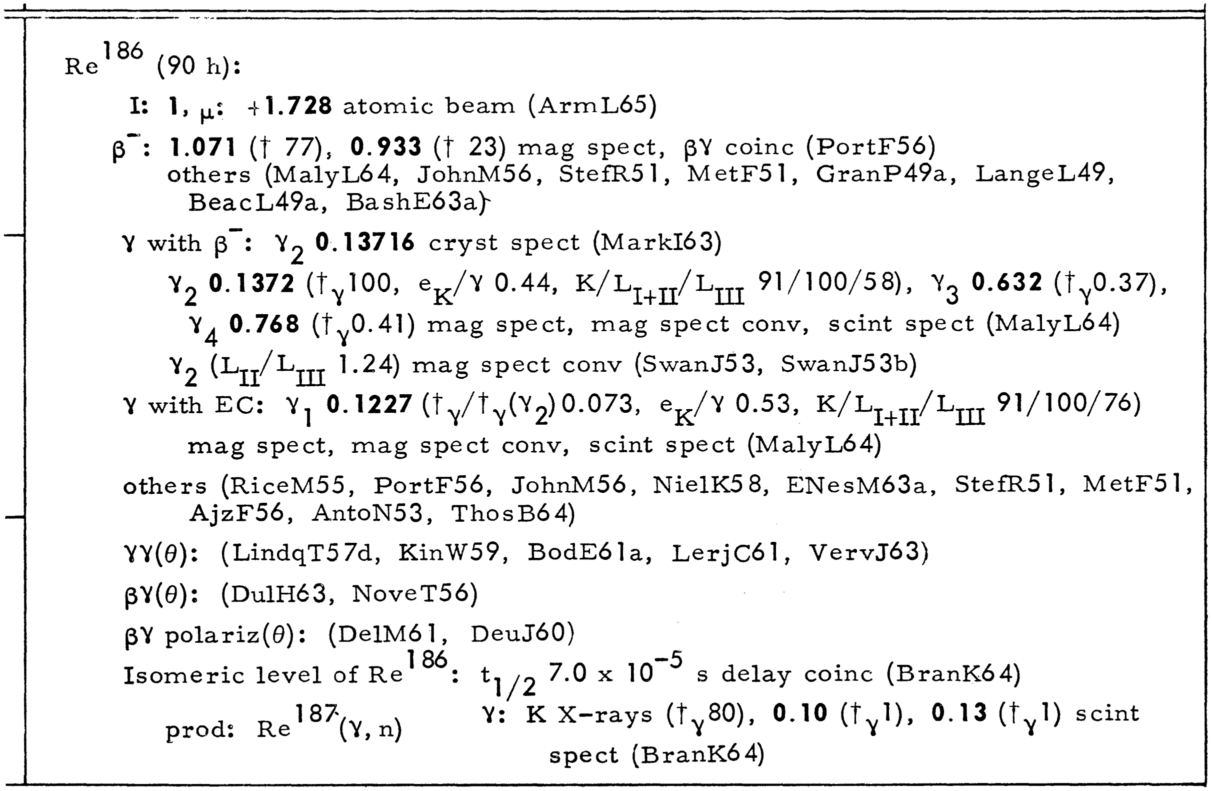
AjzF56, AntoN53, ThosB64)

$\gamma\gamma(\theta)$: (LindqT57d, KinW59, BodE61a, LerjC61, VervJ6

$\beta\gamma(\theta)$: (DulH63, NoveT56)

XBL 703-464

Fig. 5. Computer-generated microfilm plot of technical text sample.



Re^{186} (90 h):

I: **1**, μ: +**1.728** atomic beam (ArmL65)

β^-: **1.071** († 77), **0.933** († 23) mag spect, $\beta\gamma$ coinc (PortF56)
others (MalyL64, JohnM56, StefR51, MetF51, GranP49a, LangeL49, BeacL49a, BashE63a)

γ with β^-: γ_2 **0.13716** cryst spect (MarkI63)

γ_2 **0.1372** ($\dagger_\gamma$100, e_K/γ 0.44, $K/L_{I+II}/L_{III}$ 91/100/58), γ_3 **0.632** ($\dagger_\gamma$0.37), γ_4 **0.768** ($\dagger_\gamma$0.41) mag spect, mag spect conv, scint spect (MalyL64)

γ_2 (L_{II}/L_{III} 1.24) mag spect conv (SwanJ53, SwanJ53b)

γ with EC: γ_1 **0.1227** ($\dagger_\gamma/\dagger_\gamma(\gamma_2)$0.073, e_K/γ 0.53, $K/L_{I+II}/L_{III}$ 91/100/76) mag spect, mag spect conv, scint spect (MalyL64)

others (RiceM55, PortF56, JohnM56, NielK58, ENesM63a, StefR51, MetF51, AjzF56, AntoN53, ThosB64)

$\gamma\gamma(\theta)$: (LindqT57d, KinW59, BodE61a, LerjC61, VervJ63)

$\beta\gamma(\theta)$: (DulH63, NoveT56)

$\beta\gamma$ polariz(θ): (DelM61, DeuJ60)

Isomeric level of Re^{186}: $t_{1/2}$ 7.0 x 10^{-5} s delay coinc (BranK64)

prod: Re$^{187}(\gamma, n)$ γ: K X-rays ($\dagger_\gamma$80), **0.10** ($\dagger_\gamma$1), **0.13** ($\dagger_\gamma$1) scint spect (BranK64)

XBL 703-465

Fig. 6. Computer-generated CalComp plot of technical text sample.

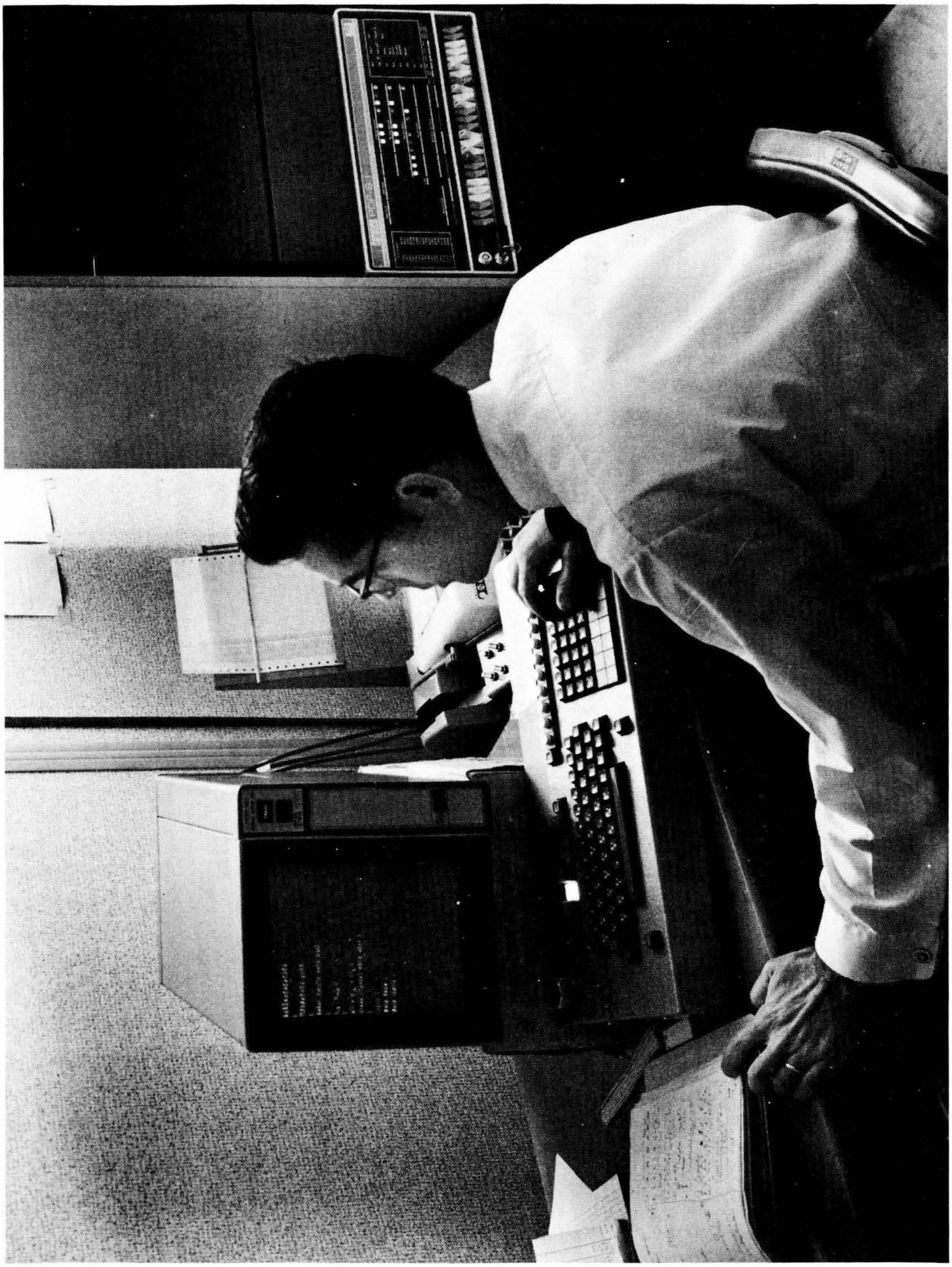
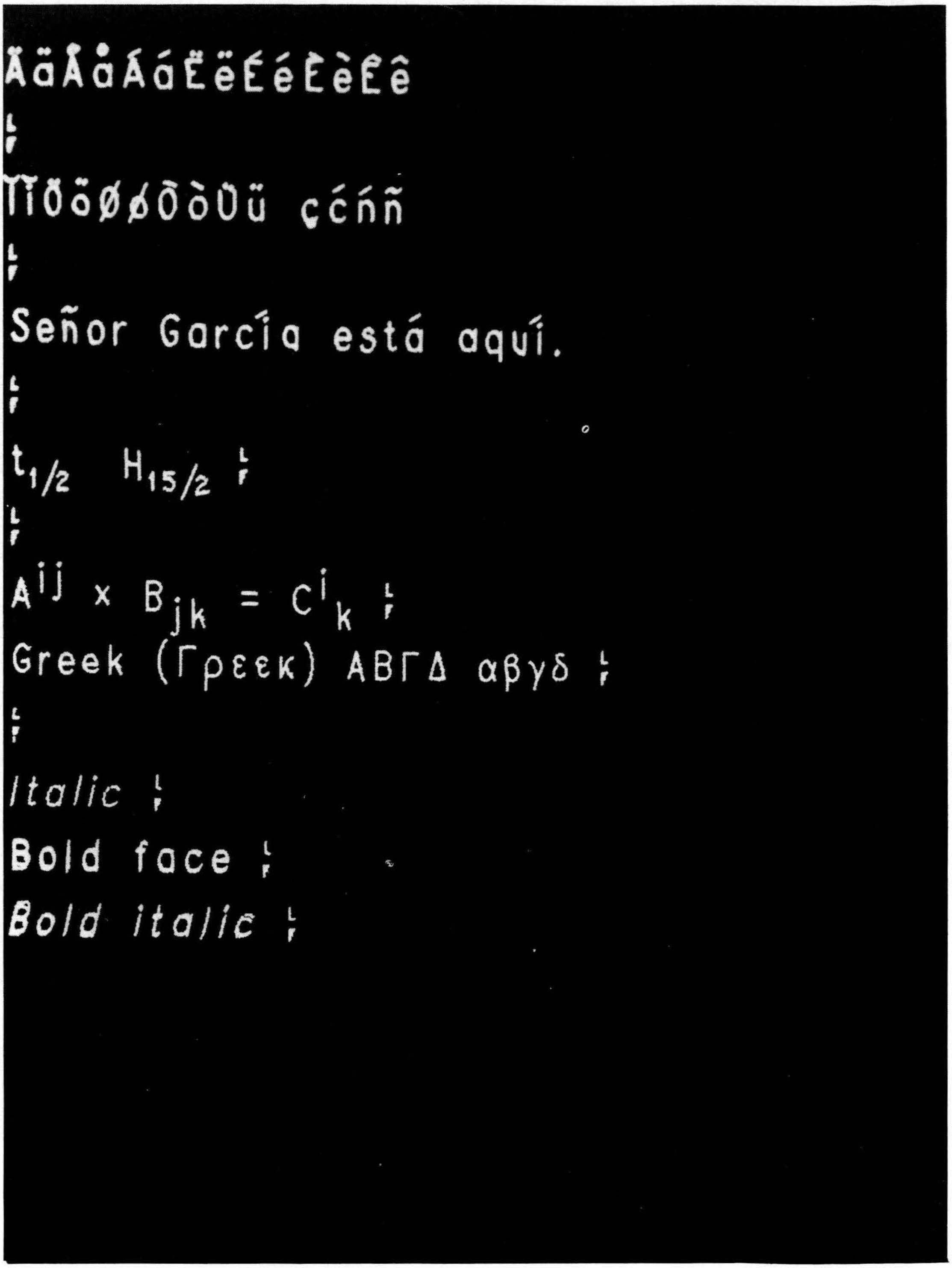


Fig. 7. On-line technical text entry terminal, including keyboard and storage CRT scope.



ÄäÅåÁáËëÉéÈèÊê
⁏
ÏïÖöØøÕõÒòÜü çćńñ
⁏
Señor García está aquí.
⁏
$t_{1/2}$ $H_{15/2}$ ⁏
⁏
$A^{ij} \times B_{jk} = C^{i}{}_{k}$ ⁏
Greek (Γρεεκ) ΑΒΓΔ αβγδ ⁏
⁏
Italic ⁏
Bold face ⁏
Bold italic ⁏

Fig. 8. String of text displayed on storage CRT scope at on-line terminal.

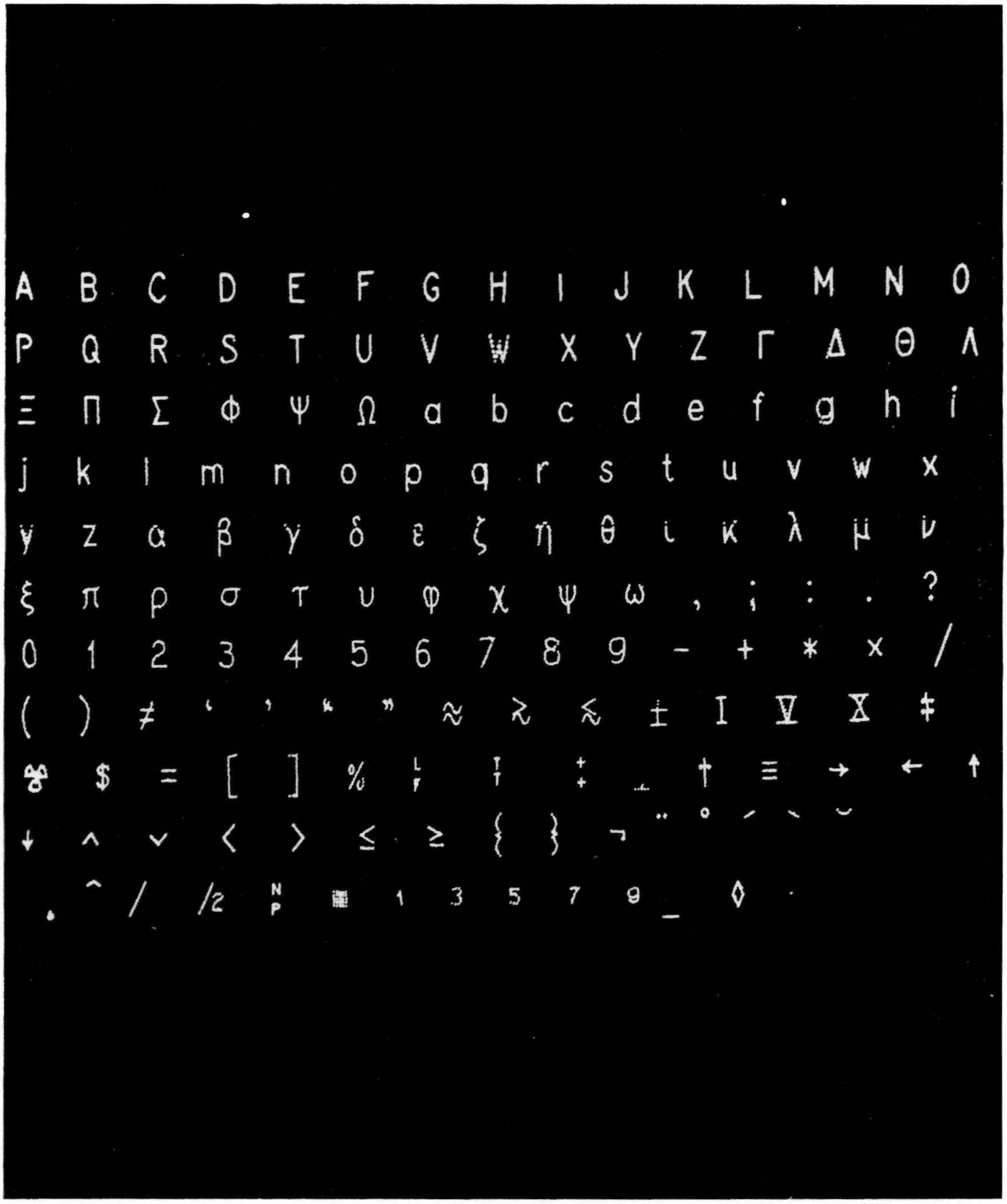

A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z Γ Δ Θ Λ
Ξ Π Σ Φ Ψ Ω a b c d e f g h i
j k l m n o p q r s t u v w x
y z α β γ δ ε ζ η θ ι κ λ μ ν
ξ π ρ σ τ υ φ χ ψ ω , ; : . ?
0 1 2 3 4 5 6 7 8 9 - + * x /
( ) ≠ ‘ ’ “ ” ≈ ± I V X ‡
$ = [ ] % † ≡ → ← ↑
↓ ∧ ∨ ⟨ ⟩ ≤ ≥ { } ¬ ¨ ° ´ ` ˘
. ^ / /2 N P 1 3 5 7 9 _ ◊ ·

Fig. 9. Character repertory of on-line terminal.

DESIGN CONSIDERATIONS FOR A REMOTE GRAPHIC COMPUTER TERMINAL

Donald C. Austin and C. Norman Winningstad, Tektronix, Inc.

SUMMARY

After a brief review of the reasons for business EDP overtaking and surpassing technical applications of computers, it is shown that technical and management applications can grow considerably due to two recent changes: (1) The advent of practical timesharing services, and (2) The availability of low cost (<$10,000) graphic remote computer terminals. The problem of the graphic remote computer terminal is defined, the general solution is stated, and a specific technique which has been employed is stated. Certain communication considerations are described, and a digital format is explained. A simplified block diagram is used to describe how two alphanumeric and three graphic modes are obtained. The speed limitations of the various modes are stated.

INTRODUCTION

The rapid expansion of computers applied to the Electronic Data Processing of the business world was possible mainly because alphanumeric input and output were sufficient and "real time" turn around in the business world could proceed at weekly or longer intervals. Application to technical problems was the original emphasis for the development of computers.[1] However, the lack of low-cost graphic input and output means, the inability to reduce the turn-around time of man-machine interaction to the order of seconds or less, and the necessity of bringing the man to the machine, has held back a similar expansion of computer application to technical and managerial problems. The recent development of timesharing techniques has allowed the starting of timesharing services, which through the ubiquitous teletypewriter, has shown the way to reasonably economic, rapid, and convenient access and turn-around time for the solution of certain technical problems.[2] The limitations of the teletype format and speed have, up to very recently, been a serious limitation to general use of timesharing for a broader range of technical and managerial problems.

The Problem

Graphic terminals, for other than trivial or special purpose uses, require a considerably higher addressability and resolution capability than alphanumeric (A/N) only terminals, and generally has a need for higher total information capability in the display. The desirable remote location of graphic terminals (bring the machine to the man) then requires (for economic reasons) relatively narrow bandpass communication lines. Since the impulse speed of human vision is of the order of one-tenth of a second, in order to see a whole display continuously, some form of memory must be at the remote location, which will accumulate the data sent from a computer through the relatively slow communication lines. This remote memory can be a property of the display, or a separate electronic memory, which refreshes an ordinary, non-storage display. The economics suggest that for relatively low resolution A/N displays, low-cost memories can be used to refresh ordinary cathode ray tubes (CRT) to do the job. Figure 1 shows a use domain composed of "Total Resolution", as an ordinate, in terms of displayable A/N, vs "Refresh Rate" in hertz. Note quality refreshed displays must operate to the right of the flicker fusion zone.[3] Thus as the amount of stored information goes up, an electronic memory must not only be larger, but also be faster. Another problem is the CRT system. As the information to be displayed goes up, the CRT system must also become not only faster, but also brighter, since the duty factor of a given point on the screen is decreasing. Thus today in simple A/N applications where the teletype is too slow, or noisy, one finds inexpensive acoustic delay lines refreshing simple CRT's. As the number of A/N increases, one must change to more expensive memories, such as core, and make use of the economy of scale by grouping several terminals around one memory, and spend considerably more on the CRT display. Finally, if one wishes to use that system for graphics or very high numbers of A/N, one finds that even systems costing over $100,000 must dip below the flicker fusion rate* because of the change-to-mass ratio of the electron limiting the state-of-the-art in CRT systems. Thus for high resolution A/N or graphics, a refreshed system is under an economic handicap, not only in the refreshing memory, but also in the display, because of the high display rate imposed by the flicker fusion zone.

The Solution

By incorporating the storage requirement as an integral part of the display, rather than as an electronic memory separate from the display, the problem of speed is greatly reduced.[4] It is now only necessary for the memory display system to be fast enough to stay ahead of the limited bandwidth communication system. Although this relaxed requirement on speed requires a sacrifice of the possibility of real time displays (see Figure 1.); in fact it is not a serious disadvantage, since: (1) Most problem-solving is done in stop motion, and (2) The limited communication bandwidth precludes real-time on a refreshed display also (excepting the case employing a local computer to provide change information, in which case one is no longer dealing with a "remote" computer terminal). There is the further advantage of no flicker, even when the whole screen is filled with information.

---

*IBM Type 2250

A Particular Example

The simplified bistable storage CRT (SBSCRT)[5] is an example of an approach to the combining of the display. This particular approach is advantageous in the case where resolution on the order of 1024 by 1024 grids is desirable (at somewhat lower resolution, storage matrix displays may in the future become competitive when they are a reality, and the economics of the driver improves). The one-man one-terminal relationship is a fortunate one, since it allows close-up viewing; thus the economics are further served by providing the required resolution on a SBSCRT of modest dimensions and brightness. For group viewing, where large screens and brightnesses are necessary, a scan conversion[6] can be used in conjunction with the SBSCRT to convert to a television display.

Communication Considerations

Since the most readily available communications lines today are the telephone system lines, they will most likely form the basis of most all of our timesharing networks. There are a number of devices (modems[7]) available to optimize the transmission of digital data over the 'phone lines. At the present time, ordinary lines can be used with good reliability to 1200 baud. Faster rates than this cause the price of the modem to rise rapidly. Without becoming deeply imbroiled in this subject, it would appear that full duplex is more useful in a graphics terminal than in a teletype terminal, and if the channels must have differing rates, the computer-to-terminal should be faster than the terminal-to-computer (based upon human typing capabilities). Early efforts seem to be following the teletype code format, expanding in most cases to include the full United States of America Standard Code for Information Interchange[8] (USASCII or ASCII), and employing various types of coding for graphics.

A general view first may be helpful. As usual, a compromise may be made between software and hardware. For example, a terminal could operate entirely in a simple "point-plot mode", this means addressing the beam of the SBSCRT to each and every point separately to form a given A/N or graphic message. On a 1024 by 1024 grid, this would require transmission of a 10-bit X address and a 10-bit Y address, and this would result in a point being plotted. Thus with proper software, the A/N or graphic can be traced out. This makes the terminal quite inexpensive, since no character generator is needed, and the CRT addressing is simple; however, communication time is long. A step more sophisticated at the terminal is to include a stroke generator (usually a unit-sized vector available in from 8 to 256 or more directions). By keeping the unit stroke short, all the A/N and graphics can be accomplished with good fidelity, but at the cost of many instructions just to draw a single long line. The software is now simpler, and communications times reduced. A still more sophisticated terminal might contain a full ASCII character generator, as well as point-plot, and could in addition have special modes, such as incremental

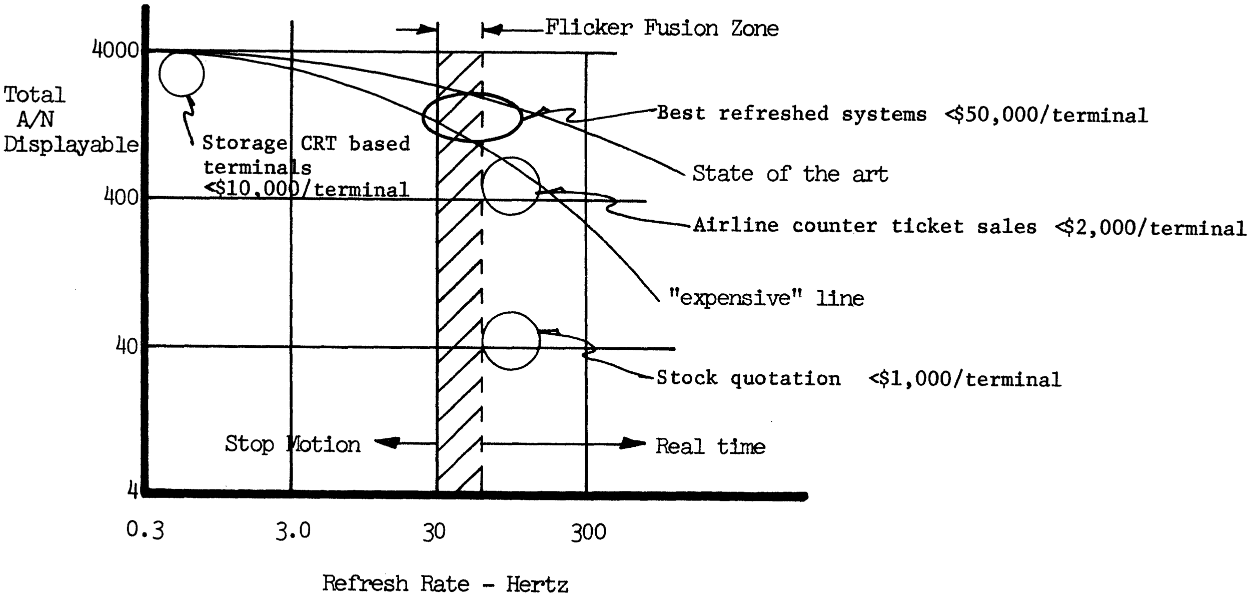


Figure No. 1

plot (very popular in "on-site", direct wired-to-the-computer applications; but incremental systems are very vulnerable to noise, in that one false-placed point will cause a dislocation in the subsequent plot which makes it very awkward to use, whereas, one misplaced point or line seldom renders a total graph worthless, as in Figure 2.).

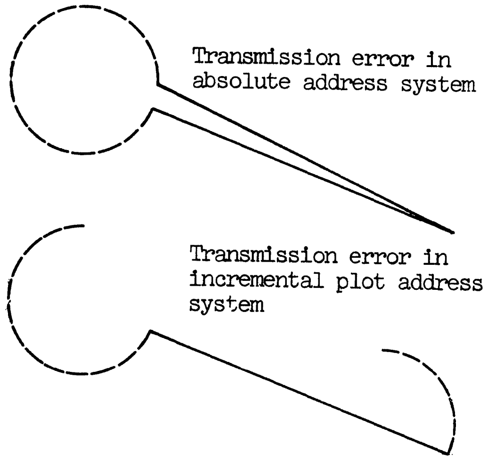

Transmission error in absolute address system
Transmission error in incremental plot address system

Figure No. 2

Another more useful graphic mode is one wherein a starting point is specified and a final point is specified; then the terminal draws a straight line between the points (called "linear interpolation"; it is possible, at the cost of complications to the hardware and software, to have "non-linear interpolation", and generate curved lines between two points). A subtle point about linear interpolation: Note it does not suffer from the quantizing effects of the D/A converter in drawing a line. It is smooth and continuous, thus avoiding moire and other stepping effects. Obviously all modes are useful, and some are faster than others, depending upon the nature of the plot. For example, text is fastest with a character generator, grid lines for a scale on a graph are faster with the linear interpolator, incremental-plot is fastest on a rapidly changing curve, and point-plot is needed for scattered data points.

For today's needs, the most sensible (economic) balance is the hardware/software/communications domain seems to be to include a hardware character generator, point-plot, incremental plot, and linear interpolate capabilities; this is largely because communications costs are high, and hardware costs are coming down. A 1200/110 baud full duplex modem is what the terminal user seems to prefer, but if the computer is far enough away as to require long distance lines to be used, the time-sharing service would appear to prefer 300/300 baud as a workable compromise; this will be likely true as long as the time-share services parallel many terminals' data by frequency division of a channel; but if they change to time division, then high bit rates would be favored. Since time division makes better use of the spectrum than frequency division ("guard-bands" require more frequency space than "identifiers" require time-space), the change is inevitable as high speed modems come down in price.

Digital Format

The remote graphic computer terminal to be described is for a version utilizing an EIA Standard RS232 compatible communications interface.[9] The coding on the serial data lead has the standard teletype codings of ten or eleven bits; they are in sequence a start bit, seven data bits, a parity bit, and one or two stop bits. The data rate may vary from less than the normal teletype rate (110 bits/sec) to as high as 50,000 bits/sec. The seven data bits are used in different ways, depending upon the terminal mode.

There are two A/N modes and three graphic modes. The first of the A/N modes is the ASCII character set (96 "printing" and 32 non-prints commands). The second A/N mode, selected by a push button, is referred to as "TTY", since it converts the full set standard keyboard to automatically act like a teletype keyboard; i.e., it normally sends only capital alphas, and is extremely convenient in that no keyboard shifting is required to go back and forth between capital alphas and numerals. The change from full ASCII to TTY is within the keyboard of the terminal only, and the same eleven-bit data string simply appears with the seven data bits of the ASCII set in the output (or input) in either mode, in other words, the TTY code is just a sub-set of ASCII. The terminal normally starts up in the ASCII A/N mode, and if in a graphic mode, is returned to whichever A/N mode was previously selected by any one of several means: Control characters "carriage return" (CR) and "unit separator" (US), or "Home" button command.

In both A/N modes, spacing between letters proceeds automatically left-to-right, and spacing between lines proceeds automatically top-to-bottom, with an electronic bell warning eight and five characters before the end of the 85 character capacity line and a "page full" warning light a patchable number of lines before the 39 line page capacity is exceeded. The location of the start of an A/N sequence is addressable to any feasible point within a 1024 by 742 grid, which is different from the usual "pigeon hole" format of A/N only terminals.

In the graphic modes (which may be utilized to address the starting point of an A/N sequence) there is a 1024 by 760 point grid available. The first of the three graphic modes, incremental plot, has available a one grid-point movement, from a previous address, in any of eight directions corresponding to one step up or down, left or right, or combination thereof, and a decision to write the point or not. The eight unit incremental motions are numbered for convenience "0" for vertically up, clockwise around to "7" for 45 degrees up to the left. By sending the control character "Record Separator" (RS), the terminal is set up to operate in the incremental mode.

The next non-control ASCII character will be interpreted as an incremental command. The three least significant bits (left three in this seven data bit format) are coded for the binary 0 through 7 for direction selection. The next bit is a zero for a write command and a one for a "don't write" command, the fifth bit is ignored, and the 6 and 7 bit can be anything except two zero's (since that is a characteristic of a control character). Since there are six possible arrangements of the three right-hand data bits, which do not violate the "no control character" rule, a vertically upward increment could be written by an ASCII "0" (normally used), or any of the following: @,\,p,P, or a "space" character. Similarily, there are six ways each to specify the other directions "write", as well as "don't write".

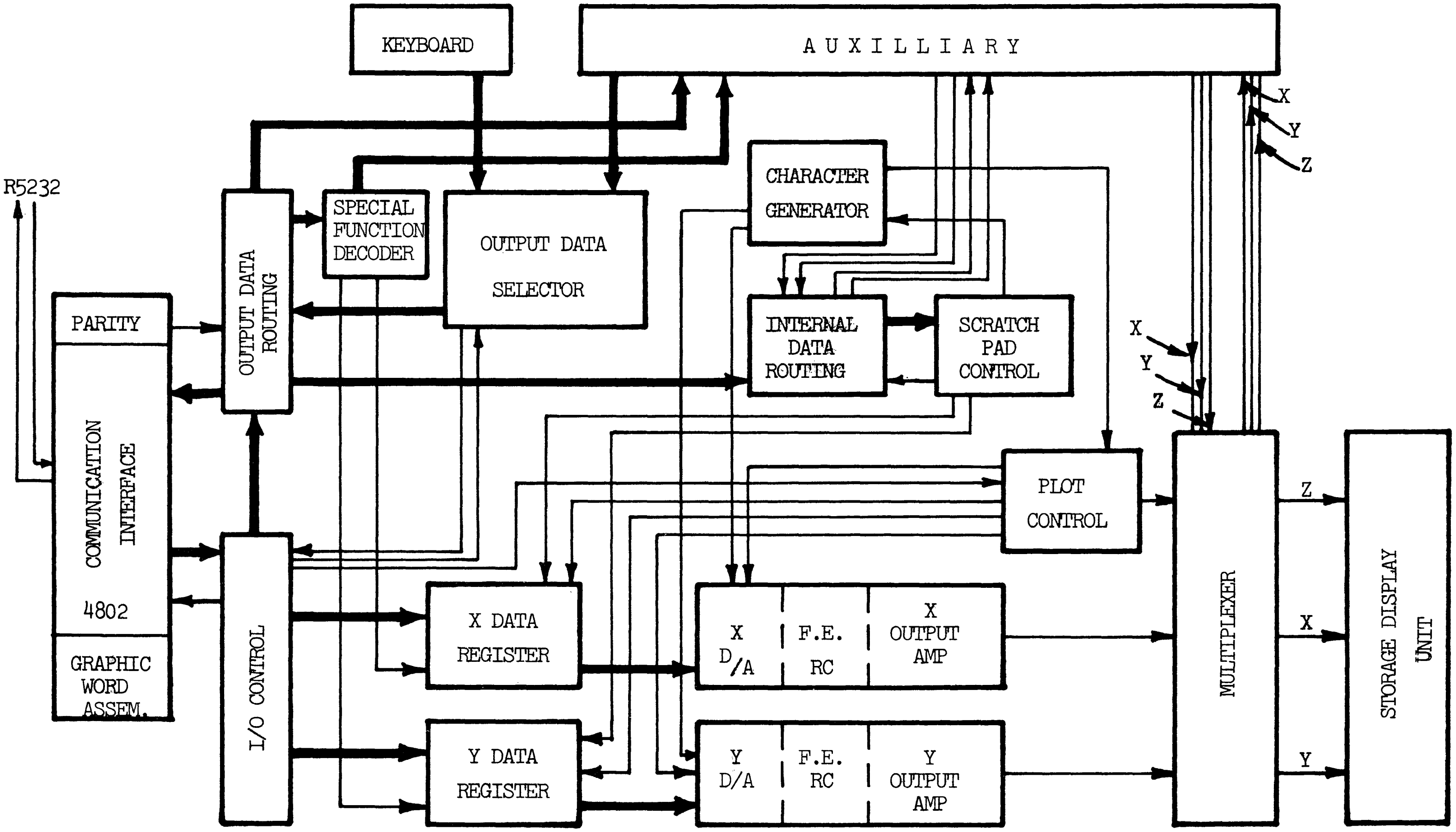
Both the A/N modes and the incremental-plot mode make small movements on the SBSCRT display, and so grid address changes are small. However, the point-plot and linear-interpolate graphic modes have the possibility of very large address changes. Since there are ten bits each required for a complete X and Y address, and only seven data bits available in the ASCII format chosen, a "graphic word assembler" is used to allow a complete addressability. When the control character "File Separator" (FS) or "Group Separator" (GS) is sent, signifying point-plot and linear-interpolate respectively, the terminal knows it must now employ the Graphic Word Assembler (GWA). The left-hand 5-bits of the ASCII word, when in GWA mode, is considered to be data bits. The right-hand 2-bits are identifiers. A "01" identifier following a control character (FS or GS) is interpreted as signifying the five data bits are the "Y" high order bits. A "11" signifies the five data bits are low order Y bits. A "01" following a "11" character signifies the five data bits are high order "X" bits; a "10" signifies low order X bits, and causes the "Execute" command to the terminal graphics, and if the next identifier is a 01, causes it to be high order Y data.

This simple coding scheme allows rapid transfer back and forth between the various modes, as well as minimizing the number of 11-bit sequences necessary to move about the screen. The point-plot mode obviously makes use of absolute addresses (as opposed to incremental plot, where one moves a certain distance from the last point). Linear interpolate plotting is a bit different, in that the first address after receipt of the control character GS causes a non-writing traverse from the last received point to the new point. Then the next point received causes a written line to appear from the end of the non-writing traverse to the new point. The use of absolute addresses in the linear interpolate mode is considered very important to minimize the consequences of an occasional transmission error: e.g., circles will close, and although it might contain a flaw in the circumferential path (typically in the form of a two-line segment path connecting a break in the circle), one knows the circle quite well, and the eye easily ignores the flaw (see Figure 2.).

The location of the beam of the SBSCRT is signified by a cursor, whose position is influenced by the two A/N modes and the three graphic modes, as well as the "Home" (moves to the upper left corner), left-right, and up-down buttons; additionally, certain control characters (such as "Start of Heading", or "SOH", which acts similarly to "Home") will also move the beam in standard ways. Control characters which are of built-in significance to the terminal are patchable, in case the operating system of the computer should be changed, necessitating a different control character choice.

Block Diagram

Figure 3 is a simplified block diagram of the terminal. Assume the keyboard is selected for an input. Upon striking a key, the ASCII coded signal proceeds as 8-wire parallel to the communications interface. Here the signal is converted to serial bits, is sent to the computer via a modem, and if the computer system is capable of echoplexing, the character is returned (or if the computer does not echoplex, such as half-duplex systems, then the communications interface is switched to provide a local echoplex). The echoed character is parity checked (in case of parity error the character is displayed in italics and an error light is lit), and sent in parallel to Input-Output (I/O) Control (all data bits are parallel within the terminal except for the Scratch Pad-to-Auxillary, which is bit-serial). The routing through Output Data Routing (ODR) and Internal Data Routing (IDR) ends up at the Character Generator (CG). The CG is basically a 9 x 7 dot matrix, which is traced through serially at about 0.5 µsec per unwritten dot, spending about 5 µseconds to settle at each written position and 5 µseconds to write. The maximum number of written dots in a character is in the letter "B", which is 30; the minimum is the ".", which has 5. The highest rate with an "average" character of 17 dots is about 5,000 A/N per second; as the CG scans through its matrix, the CG sends position information to the X and Y Digital to Analog (D/A) converters, causing the display to track the CG scanning with respect to a position specified by the last data in the X and Y Data Registers (DR). The 9 x 7 raster is traced with one addressable computer grid per raster step. (The inter-character space is approximately five steps, and the interline space is about 10 steps.) The resolution is approximately 70% of the addressibility, so diagonally related portions of an A/N (or graphic) are about tangentially displayed. A point is written by the CG Read Only Memory detecting the points within the 9 x 7 raster which should be displayed. This information is sent to Plot Control in order to command the display device selected by the Multiplexor to write, and to properly step the X DR for the next A/N; the Y D/A is commanded by the CG to accept an offset for certain characters, so that the tails of "y", "g", etc., will appear below the line. If a parity error was detected, or if a "Shift In" (SI) control character commands it, Internal Data Routing will cause the X D/A to get a ramped offset, resulting in an italicized character.


KEYBOARD
A U X I L L I A R Y
X
Y
Z
R5232
SPECIAL FUNCTION DECODER
OUTPUT DATA SELECTOR
CHARACTER GENERATOR
OUTPUT DATA ROUTING
PARITY
INTERNAL DATA ROUTING
SCRATCH PAD CONTROL
X
Y
Z
COMMUNICATION INTERFACE
4802
GRAPHIC WORD ASSEM.
PLOT CONTROL
Z
X
Y
I/O CONTROL
X DATA REGISTER
X D/A
F.E. RC
X OUTPUT AMP
Y DATA REGISTER
Y D/A
F.E. RC
Y OUTPUT AMP
MULTIPLEXER
STORAGE DISPLAY UNIT

BLOCK DIAGRAM - T4002

Figure No. 3

The X DR has a detector for the transition from the 76th to the 77th A/N position, and from 80 to 81, causing a bell to sound if in the A/N mode. The Y DR has a patchable detector to actuate the "Page Full" alarm in time to flag the computer before the Y DR is full.

Upon the Communication Interface receiving the control character FS or GS, the Communication Interface's GWA will generate a new set of high and/or low order X and Y data bits, as is appropriate, for application to the X and Y DR's. The receipt of GS also causes the field-effect transistor switches to operate an RC network between the X and Y D/A'l, and the X and Y output amplifiers respectively. This causes the output of the D/A's to drive the X and Y output amplifiers from their original valve, to a new valve, at an exponentially decaying rate. If the X and Y time-constants are matched, the spot trajectory is a straight line.[10] By controlling the Z axis unblank duty factor, a reasonably uniform stored line results.

Auxillary Interface

The auxillary interface serves two major purposes; firstly, it allows the connection of output devices other than the built-in SBSCRT, and secondly, it allows input devices other than the built-in keyboard to function. An example of another output device would be a scan converter[11], which allows group display by means of television repeaters. An example of another input device would be a "graphic input" which can be any one of a number of devices. The requirement is that they provide the X and Y address digitally to the auxillary input. This guarantees there will not be paralax between what is displayed and that which the computer receives (some systems utilize the DC voltages from graphic input devices to directly effect the display; then an A/D converter sends the digitized information to the computer, allowing a possible offset and/or scaling error to exist between the display and the digital information).

Scratch Pad

The Scratch Pad (SP) provides an 84-character buffer which can be made to display its contents in a refreshed fashion in a special non-store area at a Y extreme of the SBSCRT, or read the contents out once in the active area of the SBSCRT, or (if desired) transmitting the contents out to the computer. The buffer includes a simple teletype style editing arrangement utilizing the back space to rub out the previous character. A cursor keeps track of the present writing position. A useful feature of the Scratch Pad is that it will hold the non-writing ASCII control characters (but using up a character space in the buffer). This allows very rapid sign-on with a timeshare system, since many of the "hand shaking" and line feed requirements can be placed in the Scratch Pad. Then with a single press of a control button, a sequence of events occurs (this feature will presumably become increasingly useful when time-division of telephone lines is more common).

Speed Limitations

The SBSCRT in the terminal has a quality area of approximately 16 x 20 cm. The beam deflects to within one spot diameter in (5 plus 3½ times the X or Y deflection distance in centimeters) microseconds, and requires about five microseconds to write a spot. Thus in the A/N modes, to be sure of writing the most complicated ASCII character, the input serial bit rate is limited by the CRT writing and deflection times to about 33 kilobits/sec, or about 3,000 characters/sec.
Incremental plot is limited to a 50 kilobit rate by the Communications Interface, or about 5 kilo-points/sec. Point plotting can also go at the CI maximum rate of 50 K bits per second, or 1.25 to 5 kilo-points per second limited by 1 to 4 words/point, depending upon whether or not the GWA must change the high order bits. Linear interpolate is limited by the linear writing rate of the SBSCRT, and the choice of the RC time constant for the longest vector, to about 125 vectors/sec; at 1 to 4 words per vector, corresponds to approximately 1250 bits/sec to 5,000 bits/sec.

Hard Copy

Although one will not usually wish to have Hard Copy of what's on the SBSCRT, there will be conclusion points when a Hard Copy would be useful, such as program listing, the final results of a problem, and even perhaps the input data from which the results were obtained. Although most computer systems now provide a Hard Copy capability, you usually get more Hard Copy than you want, and at a time much later than you wanted it. A one-half scale Poloroid photo is easily taken[12], and is many times sufficiently useful, even though it is small, hard to write on, and positive (white writing on a black background). Provision has been made for a scan converter arrangement, which will provide an 8½ X 11 inch negative copy. It is interesting to note that the photograph will contain the small linearity errors of the display, but the scan conversion first-order cancels out these errors.

Software

A Software package has been made which contains diagnostic routines, and first-level graphic routines in some common languages: Basic, Focal, and Fortran[13].

REFERENCES

1. Book reference:

   Parkhill, Douglas F. (Addison Wesley) "The Challenge of The Computer Utility", 1966.

2. Journal reference:

   AFIPS, "Conference Proceedings", Vol. XXIX, 1966, "Time-Sharing Processors and Executive Systems"; also Reference 1.

3. Book reference:

   Graham, G. H. (John Wiley & Sons) "Vision and Visual Preception", 1965.

4. Book reference:

   Culler, G. J. and Fried, B. D., "The TRW Two-Station On-Line Scientific Computer", (Spartan Books, Inc.) Symposium on Computer Augmentation of Human Reasoning, 1965.

5. Journal reference:

   Frankland, Roger, "Developments in Direct View Distable Storage Display Tubes", Tektronix, Inc.

6. Journal reference:

   Edens, George and Gibson, Charles, "The Bistable Storage Cathode Ray Tube as a Scan Converter", (Western Periodicals Co.) 11th National Symposium on Information Display, 1970.

7. Journal reference:

   Department of the Navy, Electronic Systems Command, "Principles of Modems", NAVSHIPS 0967-291-6010.

8. Journal reference:

   U.S. of A. Standards Institute, New York, Univ. Decimal Class. 681.3.

9. Journal reference:

   E.I.A., Washington D.C., "EIA Standard RS232B".

10. Journal reference:

    Colpits, Robert, Allen, Dan and Vos, Tom, "Graphic Output for the Computing Calculator" (Hewlett-Packard Journal), Sept. 1969, Page 4.

11. Journal reference:

    Tektronix Catalogue No. 29

12. Journal reference:

    Tektronix Catalogue No. 29

13. Journal reference:

    Tektronix Brochure A2420, "Supporting the T4002 in a Time-Sharing Environment".

HARD COPY FROM A DIRECT VIEW STORAGE TUBE REMOTE DISPLAY TERMINAL*

A. Vezza, Project MAC, Massachusetts Institute of Technology

Design criteria of an experimental unit used to obtain hard copy from a Direct View Storage Tube remote computer display terminal for use on the DDD network are discussed. The unit and its operating environment are described, together with the results obtained using three dry process papers.

INTRODUCTION

During the last several years, a half dozen or so devices have reached the market place that make full-size, 8½ x 11 inch, human-readable hard copy of computer output from CRT displays. Typically a wideband (500 kilo-hertz or more) channel between the device and the computer is required. This has, for all practical purposes, precluded their use at sites remote from the computer facility because of the high communication costs associated with such wideband channels. Paradoxically, the last several years have also seen the establishment and growth of multi-access computer time-sharing systems which support a wide, diverse, geographically distributed (compared to the older traditional concept of facility) on-line computer user community. The emergence of these time-sharing systems has been accomplished solely with the aid of impact-printing type terminals which communicate with the computer over voice-grade telephone lines (in many cases the dial network is used), and these systems, except for a few research facilities, are devoid of remote-access CRT terminals.

Recently, terminals other than those using impact printers have appeared in the market place. There are several types, some of which produce computer output on various paper media, while others use a CRT as the display medium, which by nature produces a rather transitory image compared to the printed page. Remote CRT computer display terminals fall into two categories; those that are primarily alpha-numeric in nature, and those that also allow some graphic capability extending from output only to both input and output.

Terminals possessing both alpha-numeric and graphic capabilities, although somewhat more expensive than those possessing alpha-numeric capability only, are gaining acceptance because of their generality and ability to display pictoral information. Furthermore, these terminals can operate over the Direct Distance Dial telephone network, and can receive and display information up to ten times faster than the older, more conventional impact-type teleprinters. Such terminals are finding use in inquiry-response type applications where the production of a permanent, human-readable record may be deferred until the desired results are found, and in other applications where the interim computer output has only transient value. Examples of such applications include editing of all types, engineering design, management information and control systems, information retrieval systems, and in applications where the final result is obtained by an iterative procedure. Not only have remote-access CRT computer display terminals found ready-made applications, they have encouraged the development of systems that, without them, would have at best been awkward to use. But at present, when a user of one of these CRT terminals requires a more permanent record in the form of hard copy, he must resort to some type of teleprinter, or, if

---

*Work reported herein was supported in part by Project MAC, an M.I.T. research project sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Nonr-4102(01).

graphical information must be recorded, it is accomplished at the computer site, and the record delivered by messenger or the U.S. Postal System. It should be noted that, in both these methods of obtaining a printed record, the user must in some fashion regenerate the computer output.

If, indeed, transient-type computer terminals capable of graphic display are going to proliferate, a means of obtaining a more permanent record at the terminal site is a necessity. This fact was recognized during the final stages of the development at Project MAC, in the latter part of 1967, of a remote graphic terminal, and some experiments were conducted throughout 1968 to determine if it were possible to develop a hard-copy unit that would bridge the hiatus between the more conventional remote teleprinters, which are mainly alphanumeric devices, and the types that have a high through-put capability but require a wideband connection to the computer. This paper describes the basic goals of such a device, an experimental hard-copy unit that was constructed to perform some experiments, and discusses the results of the experiments. The feasibility experiments were conducted using 3M type 777 photographic dry-silver paper,(4) Eastman Kodak Kind 1877 and Kind 1991 dry-silver papers.(2) The 3M paper is a heat-developed direct print-out kind, and the Eastman papers are heat-stabilized photodeveloped direct print-out types.

DESIGN CHARACTERISTICS OF A REMOTE-ACCESS HARD COPY UNIT

There are numerous characteristics that may be considered germane to the design of a hard copy unit. Some of these are associated with the copy itself, i.e., quality, image stability, etc. Others are associated more with the human factors area of the design. Still other characteristics are associated with the system or type of system in which the hard copy unit operates. Some important design criteria are:

1) Image quality of the copy should show negligible degradation from the image quality that can be obtained with a soft-copy terminal. In fact, a cogent argument for the desirability of superior image quality can be made on the grounds that a hard-copy unit may support many soft-copy terminals, and could therefore be used in some instances to clarify detail that may not be completely legible on the soft-copy terminal.

2) A good quality paper with a surface that would accept both pencil and ball-point pen annotations should be used.

3) The image size should allow reading of all detail available on the soft-copy terminal without the aid of a viewing device. Although viewing devices are becoming handier to use, and one that can be held in one's lap is on the market, annotating such copy must be considered awkward, if not impossible in most cases.

4) A dry-process print-out is highly desirable. One need only recall how well the dry office copier was accepted to understand the rationale for this.

5) Information and image format transmitted from the computer to the hard-copy device should be identical with that of the soft-copy terminal. If a code or image format conversion is performed in the computer, an additional computational overhead is incurred. In this case computation costs can run as high as a dollar or two per page. Furthermore, communications costs can rise astronomically.

6) Cost per copy should compare favorably with that of standard business office copier costs.

7) Image stability of the copy should be measured in years, and preferably the copy should be of archival quality.

8) The copy availability should be such that the act of requesting a hard copy should entail an interruption of the user's ability to use the soft-copy terminal of no more than a few seconds. Also, the copy should be available at the terminal site within a minute or two.

9) Any implementation should be capable of being integrated into the soft-copy terminal. From an

economic viewpoint, any substantial increase in the terminal costs should be avoided.

Needless to say, some compromises had to be made, and not all the characteristics desirable in a hard-copy unit, or the copy produced from it could be achieved. The image quality of the copy produced by the unit described herein is not quite as good as the one on the soft-copy terminal. Also, cost per copy is somewhat higher than office copier costs and the copy stability is not archival.

EXPERIMENTAL HARD-COPY UNIT

The experimental hard-copy unit developed is an adjunct to a type of remote-access alpha-numeric and graphic computer terminal. This type of terminal (5) comprises: a Tektronix 611 Direct View Storage Tube (DVST) monitor, a set of electronics which provides x, y, and z signals for deflection and intensification, along with the control signals erase, write-through, and view. The electronics provide for a keyboard for alpha-numeric input, a joystick or mouse for cursor control and graphic input, the necessary circuits for point, line, and character-drawing ability, and an interface to a communications modem; in this instance a Bell System type 202C6 Data Set was used. The unit is capable of operation at 1200 bits/second in the half-duplex mode on the DDD network. The terminal sends and receives a character asychronous serial bit stream to and from the computer. The characters are ten bits long; they include a start, stop, and parity bit along with seven bits of information interpreted according to the ASCII code conventions. See Fig. 1 USASCII Character Assignments.

The terminal is modal in form; its primary mode being character plotting. Point plotting and vector-drawing modes are entered whenever the terminal receives one of the proper ASCII control characters. Character plotting mode is re-entered in a similar fashion.

There are a variety of techniques that can be employed to obtain a more permanent written record of the information displayed on the DVST monitor. Some are conventional and presently used, otheres are not so conventional. Some examples are: (a) A conventional wide angle optical system can be employed. (b) When a wide angle optical system is needed a scanning lens can be used to produce a more uniform exposure than can be obtained with a stationary system. (c) A mosaic of non-inverting lenses can be used to produce an image. (d) A print can be made with the aid of a CRT with slit fiber optic faceplate. A video signal, obtained by scan-converting the image on the DVST, is painted a scan-line at a time as the paper is moved over the faceplate. (e) Contact prints can be obtained if a large size coherent fiber optic faceplate or a mechanical collimator is employed. This last method could employ either a two step photographic system which uses a transfer process of some kind to obtain the proper image orientation or a second DVST with the image reversed.

All the techniques mentioned above were reviewed and two, the conventional optical system and the scan-converter, appeared most promising from a technical and economical point of view. It was decided that the former method would be tried first because it appeared to be the most straightforward, and there did not appear to be any overriding economic advantage to be gained by building a unit that used scan-converters.

Several difficulties are encountered when one attempts to make a full-size one-to-one print of the image on a DVST monitor using a conventional optical system. These difficulties arise because the 611 monitor is not very bright and photosensitive materials used in dry-process recording media or in systems that produce dry copy are not very sensitive compared to the standard photograph materials that are used in conventional still or motion picture photography. If one were to categorize these materials by establishing an equivalent ASA photographic speed rating, they would be in a speed range that extended below 10^{-6} ASA up to about an ASA of 1. This is in contrast to typical color film like Eastman Kodak Kodachrome II with an ASA of 25, and Polaroid black and white film with an ASA of 3,000. Besides the poor emissivity-sensitivity combination of the DVST and dry-process media the other difficulties encountered are associated with the lens. Off-the-shelf large-aperture (f/2.5 or less) lenses which are designed for operation at unity conjugate distances, that do not introduce excessive distortions of the image and can cover a wide 11-inch diameter field of view, are non-existent. Some experimentation with several lenses yielded unsatisfactory results. To overcome the difficulty, an old but seldom-used novel imaging system was tried and found to be adequate. (Needless to say, this avoided financing a rather costly lens design or design modification.) Two identical infinity-focus Schneider Xenar f/4.5 lenses, focal length of 350 mm, and 13-inch image size in the film plane were cascaded such that the infinity conjugate sides faced each other. This combination results in (neglecting lens losses and conjugate plane separation) an f/2.25 lens with a focal length of 180 mm. See Fig. 2 System Block Diagram. The resulting system is such that the DVST image is placed in what would normally be the film plane of the lens nearest it. The light leaving this lens and entering the second one would be nearly collimated because the image would be in focus at infinity. The second lens focuses this collimated light into an image in what normally would be its film plane, which in this case is the print plane.

The paper transport used in all the experiments was a modified 3M 144 Data Recording Module. This module has a compartment at one end to house a roll of photosensitive material which is fed on to a print plane which is approximately 10½ inches long. At the end opposite the feed slot are some rollers which are used to drive the paper and a paper knife which is activated by a cam when 11 inches of paper have passed under it. The heating station consists of a heated shell with a rotating cylinder within, and associated temperature control. Typically, after the paper has been exposed, it is advanced past the knife and cut, the cut paper is fed between and in contact with the heated shell and rotating cylinder. Modifications to the 3M 144 Data Recording Module included: replacement of the mohair-covered cylinder with one of silicone with approximately 30 threads or groves per inch ground into it to prevent steam trapping.(1) Adding of insulation and rewiring the unit allowed it to operate at the higher temperatures required for stabilization of the Eastman Kodak papers.

Figure 3 DVST Experimental Hard Copy Station shows one of the displays with the experimental hard-copy device next to it. From the figure, it can be seen that the unit employs an in-line coaxial optical system with the DVST monitor mounted vertically, a lens, an electrically operated shutter, and the 3M Data Recording Module, all in that order from the top down. The unit is capable, after suitable adjustment of the heat control, of processing any of the three types of papers mentioned previously. Figure 4 Cascaded Camera Lenses shows the lens pair used and gives some indication of their size. Inside the lens cap is a six-inch steel scale.

PAPER AND PHOSPHOR CHARACTERISTICS

The pertinent characteristics of 3M 777 dry silver paper, Eastman Kodak Kind 1877 dry silver paper, and the DVST phosphor characteristics are shown in Fig. 5 Pertinent Characteristics of The Print Papers and DVST Phosphor. Eastman Kind 1877 has been replaced by an improved version, Kind 1991, "...which has superior processed image stability and similar sensitometric response."(2) Experience in using Kind 1991 shows that it is approximately three to four times faster for this particular application than Kind 1877. The spectral emissivity of the DVST phosphor is similar to that of P1 and is in the, "bandpass region", of the spectral sensitivity of both 777 and Kind 1877. The results indicate that Kind 1991 possesses a spectral sensitivity curve which makes it better suited to this application than Kind 1877.

The processing techniques of the 3M paper and the Eastman Kodak papers are different. The 3M paper is processed by placing it in contact with a heated surface at a temperature of about 125°C for a period of eight seconds. After exposure, the background of the Eastman papers is stabilized by heating the emulsion to about 240°C for a period of four seconds; and then the print is photodeveloped by irradiating the emulsion with ultra-violet light. (In the experiments this was done by hand, by holding the paper beneath two 15-watt Black Light fluorescent lamps for a period of about 20 or 30 seconds.)

EXPERIMENTS AND RESULTS

The experiments were simple; prints using various exposure times, heat development or stabilization temperatures and times, for each of the three papers were made. These prints were compared subjectively by the author and his associates.

For the 3M 777 paper, exposure times ranged from 20 to 60 seconds. The shorter exposure times were obtained at the higher development temperatures. This was accompanied by a greater background density, a loss of latitude resulting in an under-exposed -- subjectively speaking unexposed -- peripheral region. The loss of latitude is due to the fact that the higher gamma values can be achieved at the higher temperatures. Also, the decrease in intensity at the image extremities due to the lens contributed to the exposure problems at the print extremities. Figures 6 and 7 show reproductions of prints made on 3M 777 paper. These prints are black on a greyed white background.

An exposure time of 15 seconds for Eastman Kind 1877 produced a reasonable print, and 5 seconds for Kind 1991 an even better one. Longer exposure times resulted in a print in which the DVST background is markedly noticeable; shorter times resulted in loss of image at the periphery of the print. Lower stabilization temperatures are a accompanied by a higher background density at the completion of the photodevelopment and a loss of background stability. Because the gamma is constant under different processing conditions, there is no variation of the paper's latitude. Examples of prints made using Eastman Kodak Kind 1991 are shown in Figures 8 and 9. These are a deep blue on a light cream-colored background.

All three types of paper have a limited image lifetime; the background density of these papers will increase with age. In the case of 777, proximity to a radiator or heater will cause it to degrade more rapidly. For Kinds 1877 and 1991, extended exposure to light will develop out the background. However, the author has kept both 3M and Eastman papers in a notebook for over two years with no noticeable degradation of the image quality.

COMMENT

More recently the author has been informed that the Boeing Company has produced for its internal use a more sophisticated version of this type of hard-copy device, using a lens that was specifically designed for the task. The copies show no noticeable variation of the image density over the print.

SUMMARY

The characteristics of the hard-copy unit and its product may be summarized as follows:

1) Prints may contain either alphanumeric or graphic information.

2) Recordings of 4,000 characters of legible text on a 6 3/8 x 8¼ inch area are achievable.

3) Upon the completion of the transmission of the displayed information a dry-copy display may be obtained in about 20 to 30 seconds.*

4) The unit is capable of operation at a remote site using the DDD telephone network as a communications channel.

5) The copy although not archival has a useful life, when stored properly, measured in years.

REFERENCES

1. Hauck, F. D., "Thermal Processor For Silver Halide Paper," Soc. Photog. Sci. & Eng., Symposium on Photographic Processing, 65 (1969).

2. Marchant, J., "Dry Processing Silver Photographic Materials," Novel Imaging Systems Seminar, Proc., Soc. Photog. Sci. & Eng. (R. D. Murray, Ed.) 31-51 (1969).

---

*The present design is not optimal and it would be possible to reduce the waiting time to about 10 seconds.

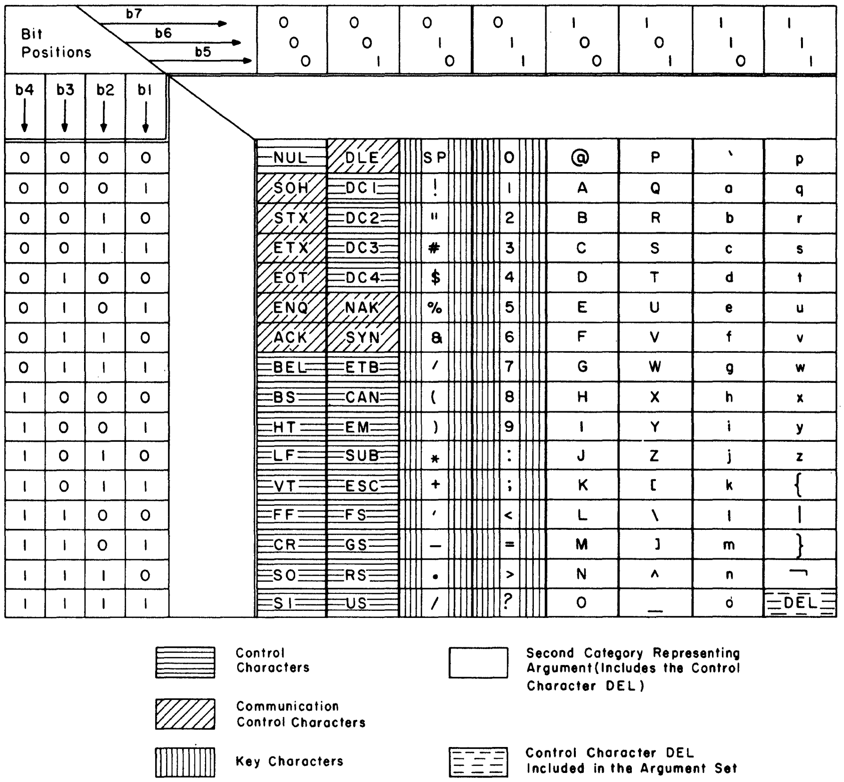
3. Morgan, D. A., Werner, T. J. and Libby, W. H., "Dry Silver Recording Materials For Display Purposes," Information Display, 44-47 (1969).

4. Morgan, D. A., Werner, T. J. and Libby, W. H., "Dry Silver Recording Materials For Display Purposes," 8th National Symposium on Information Display, Society for Information Display, 57-64 (1967).

5. Stotz, R. H. and Cheek, T. B., "A Low-Cost Graphic Display for a Computer Time-Sharing Console," 8th National Symposium on Information Display, Society for Information Display, 91-100 (1967).

6. Winningstad, C. N., "The Simplified Direct-View Bistable Storage Tube in Computer-Output Applications," 8th National Symposium on Information Display, Society for Information Display, 129-136 (1967).

7. Wurtz, J. E., "Dry Process Materials for Hard Copy Printout from CRTs," Information Display, (1969).



Bit Positions			b7	0	0	0	0	1	1	1	1
			b6	0	0	1	1	0	0	1	1
			b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1								
0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	|
1	1	0	1	CR	GS	–	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	¬
1	1	1	1	SI	US	/	?	O	_	o	DEL

▤ Control Characters

▨ Communication Control Characters

▥ Key Characters

☐ Second Category Representing Argument (Includes the Control Character DEL)

▤ Control Character DEL Included in the Argument Set

Fig. 1 USASCII Character Assignments

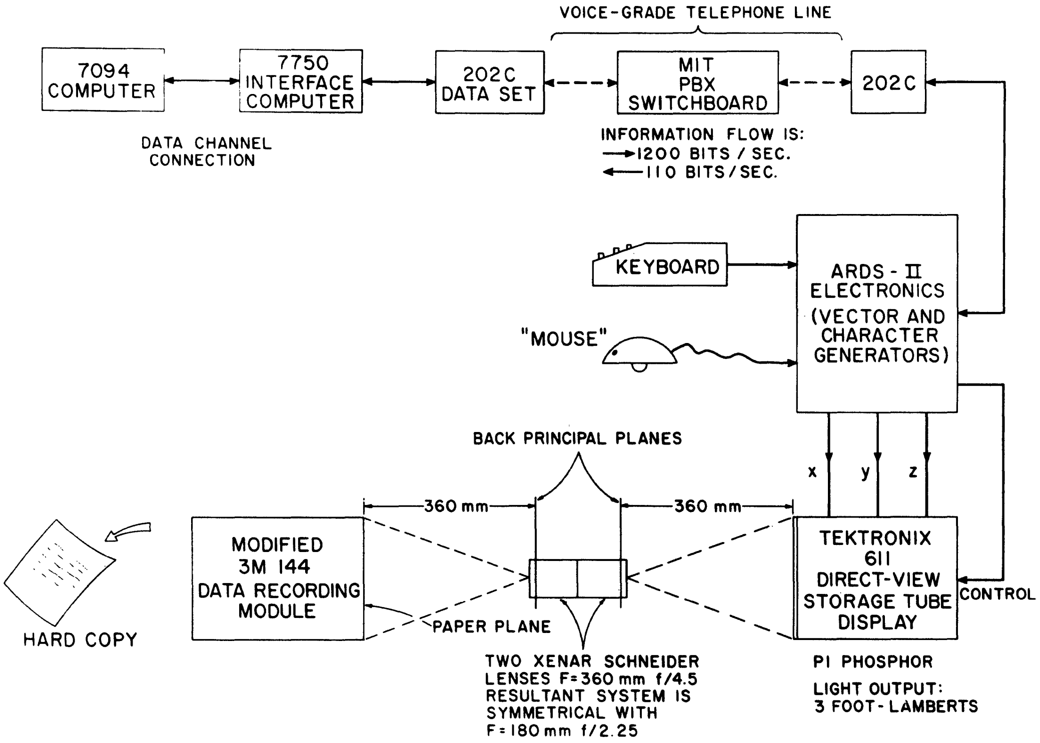

VOICE-GRADE TELEPHONE LINE
7094 COMPUTER
7750 INTERFACE COMPUTER
202C DATA SET
MIT PBX SWITCHBOARD
202C
DATA CHANNEL CONNECTION
INFORMATION FLOW IS:
→1200 BITS / SEC.
←110 BITS/SEC.
KEYBOARD
"MOUSE"
ARDS - II ELECTRONICS (VECTOR AND CHARACTER GENERATORS)
BACK PRINCIPAL PLANES
x
y
z
360 mm
360 mm
MODIFIED 3M 144 DATA RECORDING MODULE
TEKTRONIX 611 DIRECT-VIEW STORAGE TUBE DISPLAY
CONTROL
HARD COPY
PAPER PLANE
TWO XENAR SCHNEIDER LENSES F=360 mm f/4.5 RESULTANT SYSTEM IS SYMMETRICAL WITH F=180 mm f/2.25
P1 PHOSPHOR
LIGHT OUTPUT: 3 FOOT-LAMBERTS

Fig. 2 System Block Diagram

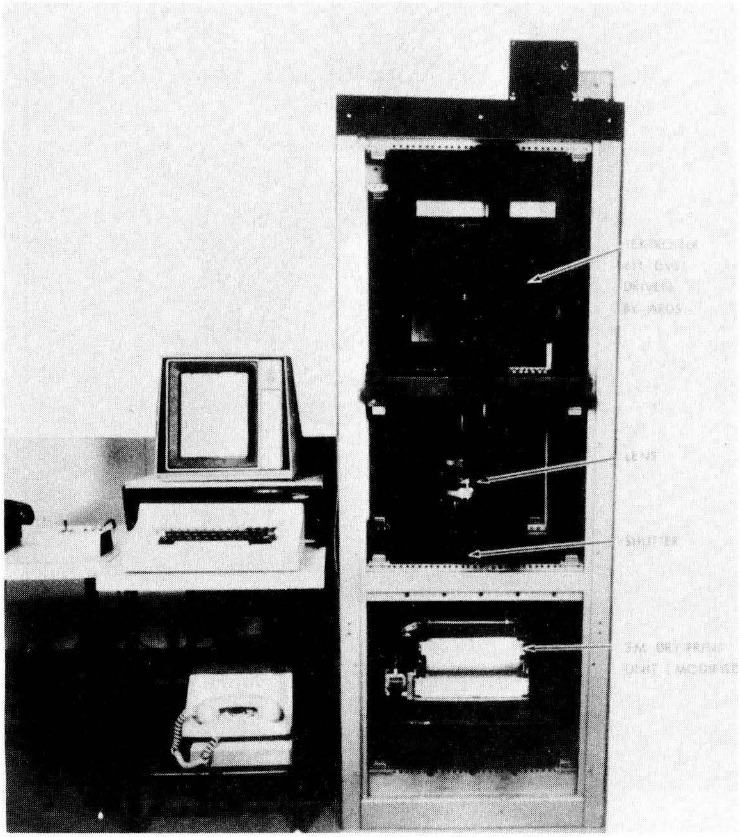

TEKTRONIX
611 DVST
DRIVEN
BY ARDS
LENS
SHUTTER
3M DRY PRINT
UNIT (MODIFIE

Fig. 3 VST Experimental Hard Copy Station

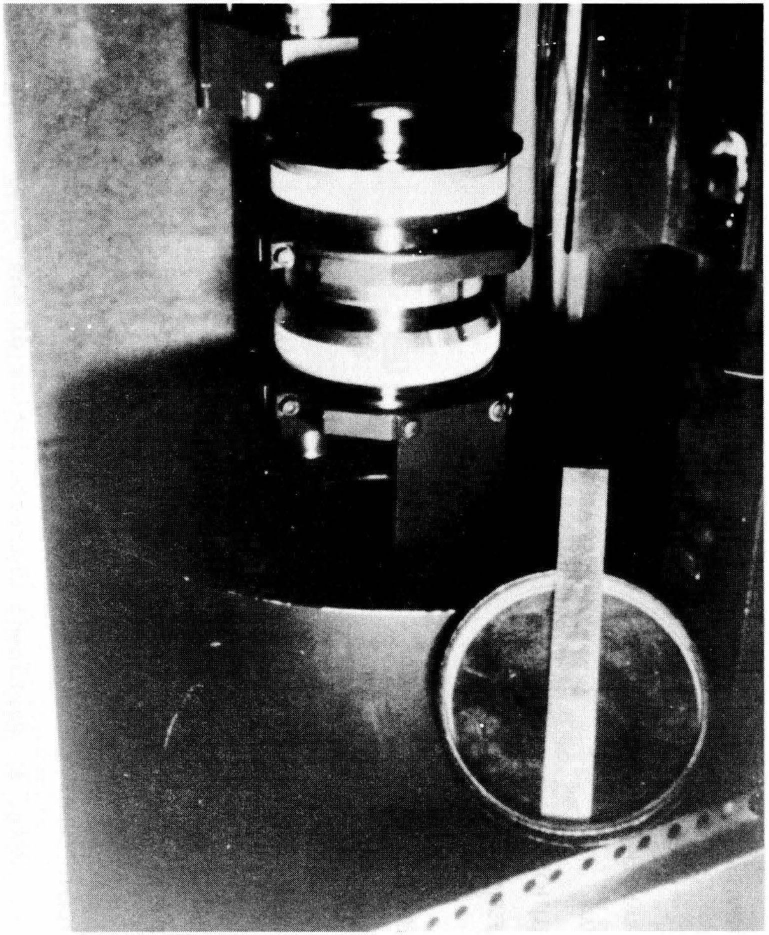


Fig. 4 Cascaded Camera Lenses

	777	D-1877
SENSITIVITY	$E_{0.6}$ ≈ 170 METER-CANDLE-SECONDS 10-100 METER-CANDLE-SECONDS	$E_{0.1}$ ≈ 3.2 ERGS/CM^2 @ 425 NANOMETERS
SPECTRAL SENSITIVITY	450-560 NANOMETERS	370-605 NANOMETERS
GAMMA	VARIABLE (1.0 to 2.4)	CONSTANT 1
RESOLUTION	EXCEEDS REQUIREMENTS	EXCEEDS REQUIREMENTS
PROCESSING	HEAT DEVELOPED	HEAT STABILIZED UV DEVELOPED
TIME	4-10 SECONDS	1-10 SECONDS
TEMPERATURE	116-135°C.	200-250°C.
UV EXPOSURE	----	1.5 x 10^4 FOOTCANDLE SECONDS TO BL FLUORESCENT LIGHT
TEKTRONIX DVST MONITOR	AT LEAST 3 FOOT LAMBERTS (~10 CANDLES/METER^2)	1.07 MILLILAMBERT
IRRADIANCE IMAGE PLANE	0.4 METER CANDLES	0.8 (ERG/CM^2)/SECONDS
SPECTRAL EMISSION P1		
PEAK	525	NANOMETERS
10 PERCENT POINTS	490	575 NANOMETERS

Fig. 5 Pertinent Characteristics of the Print Papers and DVST Phosphor

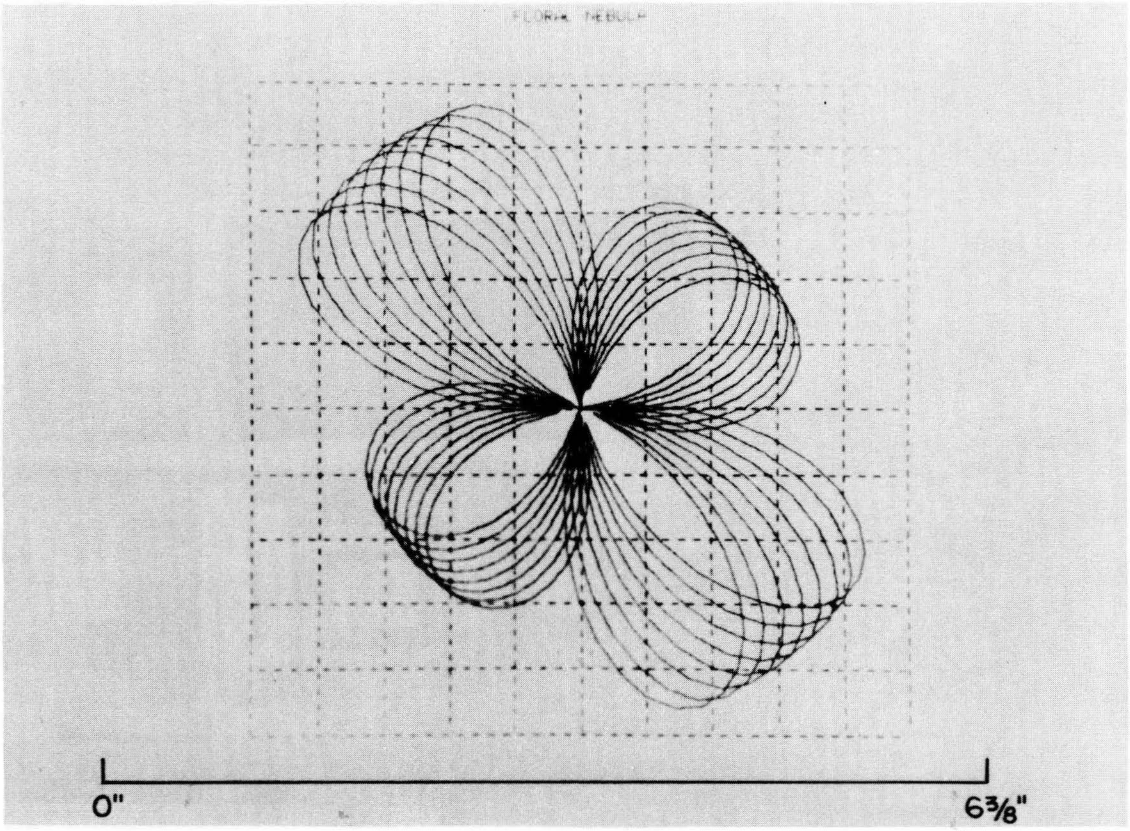
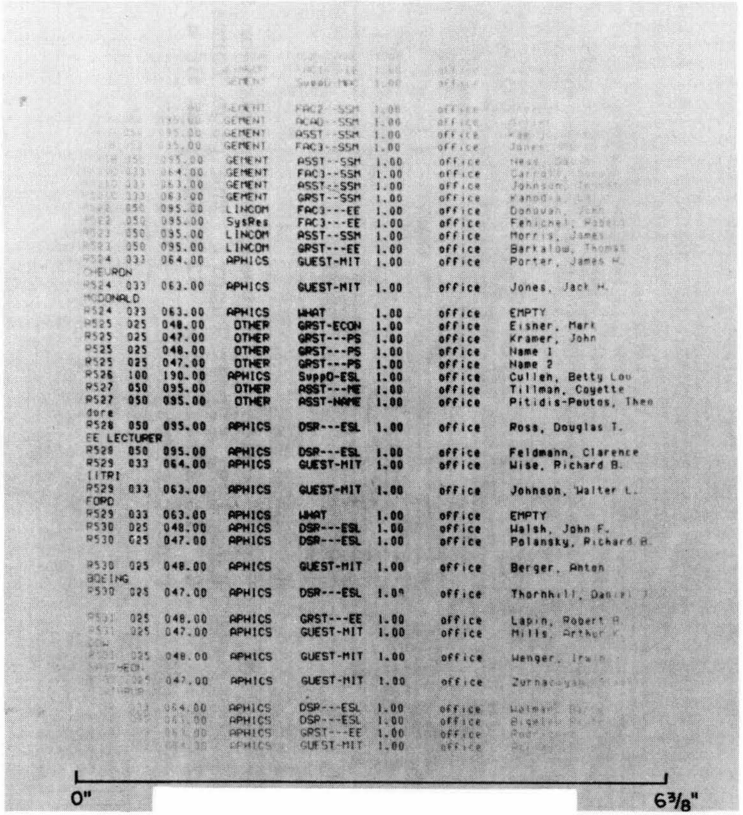
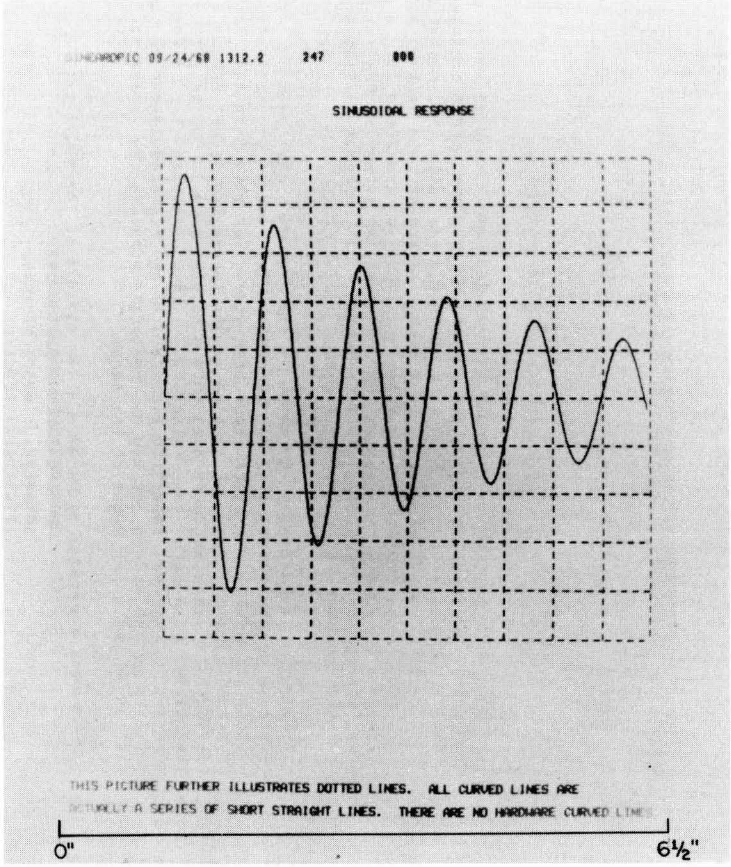


Fig. 6 Reproduction of Graphical Information on 3M 777 Dry Silver Paper



```
                    GEMENT   FAC2--SSM  1.00    office
             095.00 GEMENT   ACAD--SSM  1.00    office
             095.00 GEMENT   ASST--SSM  1.00    office   Kae
             035.00 GEMENT   FAC3--SSM  1.00    office   Jones
        050  095.00 GEMENT   ASST--SSM  1.00    office   Hess
        033  064.00 GEMENT   FAC3--SSM  1.00    office   Carroll
        033  063.00 GEMENT   ASST--SSM  1.00    office   Johnson,
        033  063.00 GEMENT   GRST--SSM  1.00    office   Kanoda
R522    050  095.00 LINCOM   FAC3---EE  1.00    office   Donovan, John
R522    050  095.00 SysRes   FAC3---EE  1.00    office   Fenichel, Robert
R523    050  095.00 LINCOM   ASST--SSM  1.00    office   Morris, James
R521    050  095.00 LINCOM   GRST---EE  1.00    office   Barkalow, Thomas
R524    033  064.00 APHICS   GUEST-MIT  1.00    office   Porter, James H.
CHEVRON
R524    033  063.00 APHICS   GUEST-MIT  1.00    office   Jones, Jack H.
MCDONALD
R524    033  063.00 APHICS   WHAT       1.00    office   EMPTY
R525    025  048.00  OTHER   GRST-ECON  1.00    office   Eisner, Mark
R525    025  047.00  OTHER   GRST---PS  1.00    office   Kramer, John
R525    025  048.00  OTHER   GRST---PS  1.00    office   Name 1
R525    025  047.00  OTHER   GRST---PS  1.00    office   Name 2
R526    100  190.00 APHICS   SuppO-ESL  1.00    office   Cullen, Betty Lou
R527    050  095.00  OTHER   ASST---ME  1.00    office   Tillman, Coyette
R527    050  095.00  OTHER   ASST-NAME  1.00    office   Pitidis-Poutos, Theo
dore
R528    050  095.00 APHICS   DSR---ESL  1.00    office   Ross, Douglas T.
EE LECTURER
R528    050  095.00 APHICS   DSR---ESL  1.00    office   Feldmann, Clarence
R529    033  064.00 APHICS   GUEST-MIT  1.00    office   Wise, Richard B.
IITRI
R529    033  063.00 APHICS   GUEST-MIT  1.00    office   Johnson, Walter L.
FORD
R529    033  063.00 APHICS   WHAT       1.00    office   EMPTY
R530    025  048.00 APHICS   DSR---ESL  1.00    office   Walsh, John F.
R530    025  047.00 APHICS   DSR---ESL  1.00    office   Polansky, Richard B.

R530    025  048.00 APHICS   GUEST-MIT  1.00    office   Berger, Anton
BOEING
R530    025  047.00 APHICS   DSR---ESL  1.0     office   Thornhill, Daniel J.

R531    025  048.00 APHICS   GRST---EE  1.00    office   Lapin, Robert A.
R531    025  047.00 APHICS   GUEST-MIT  1.00    office   Mills, Arthur K.
 
R531    025  048.00 APHICS   GUEST-MIT  1.00    office   Wenger, Irwin

        025  047.00 APHICS   GUEST-MIT  1.00    office   Zurnacoyan,

        033  064.00 APHICS   DSR---ESL  1.00    office   Wolman,
             061.00 APHICS   DSR---ESL  1.00    office   Bigelow
             061.00 APHICS   GRST---EE  1.00    office   Rod
             064.00 APHICS   GUEST-MIT  1.00    office
```

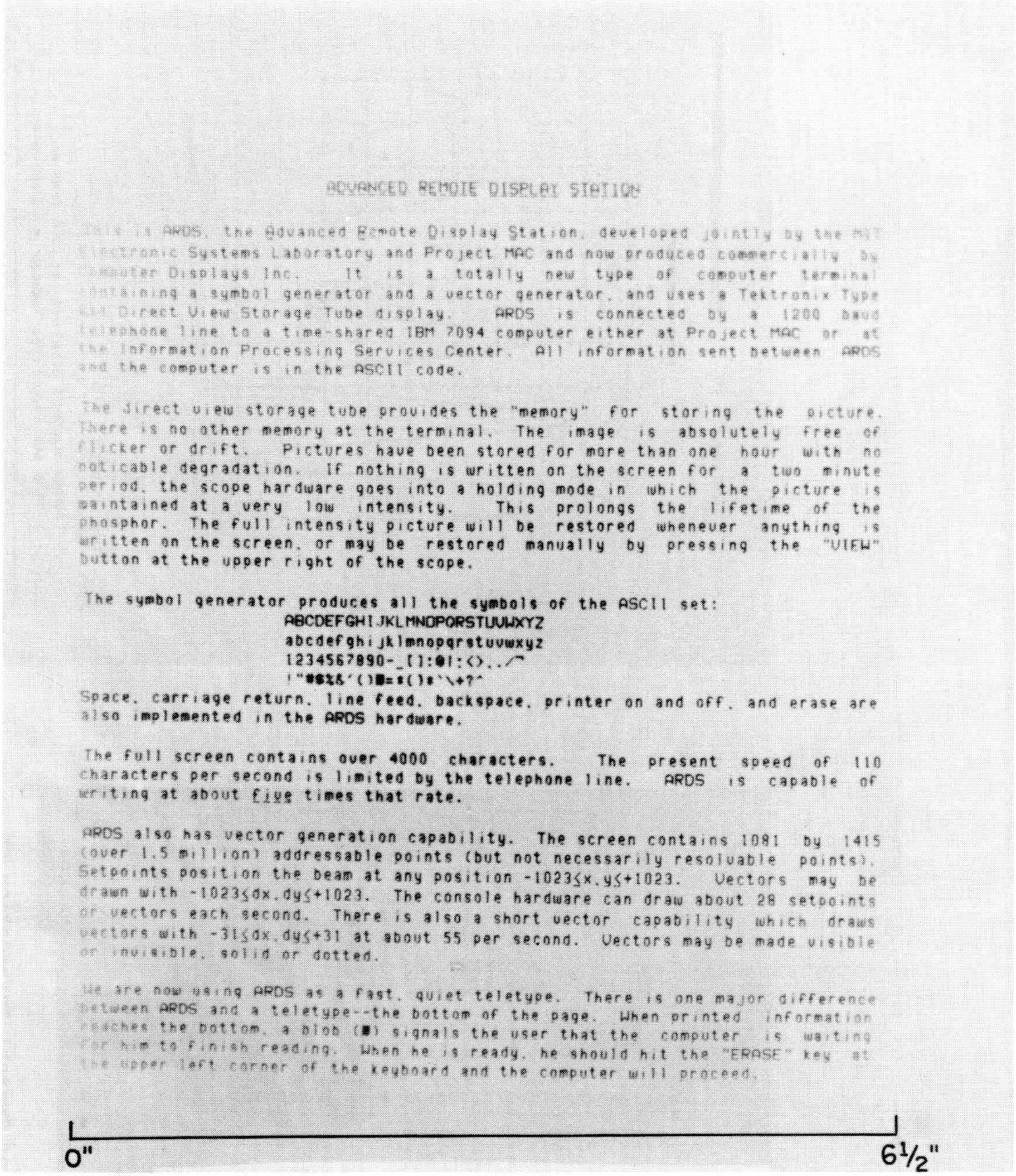
Fig. 7 Reproduction of Textual Information on 3M 777 Dry Silver paper. Page format is 80 characters/line, 54 lines/page or 4,320 character/page.



```
SINEGRAPHIC 09/24/68 1312.2     247          000

                 SINUSOIDAL RESPONSE
```

```
THIS PICTURE FURTHER ILLUSTRATES DOTTED LINES.  ALL CURVED LINES ARE
ACTUALLY A SERIES OF SHORT STRAIGHT LINES.  THERE ARE NO HARDWARE CURVED LINES
```

Fig. 8 Reproduction of Graphical Information on Eastman Kodak Kind 1991 Silver Paper.



ADVANCED REMOTE DISPLAY STATION

This is ARDS, the Advanced Remote Display Station, developed jointly by the MIT Electronic Systems Laboratory and Project MAC and now produced commercially by Computer Displays Inc. It is a totally new type of computer terminal containing a symbol generator and a vector generator, and uses a Tektronix Type 611 Direct View Storage Tube display. ARDS is connected by a 1200 baud telephone line to a time-shared IBM 7094 computer either at Project MAC or at the Information Processing Services Center. All information sent between ARDS and the computer is in the ASCII code.

The direct view storage tube provides the "memory" for storing the picture. There is no other memory at the terminal. The image is absolutely free of flicker or drift. Pictures have been stored for more than one hour with no noticable degradation. If nothing is written on the screen for a two minute period, the scope hardware goes into a holding mode in which the picture is maintained at a very low intensity. This prolongs the lifetime of the phosphor. The full intensity picture will be restored whenever anything is written on the screen, or may be restored manually by pressing the "VIEW" button at the upper right of the scope.

The symbol generator produces all the symbols of the ASCII set:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890-_[]:@!:<>../⌒
!"#$%&'()■=*()*`\+?^
```

Space, carriage return, line feed, backspace, printer on and off, and erase are also implemented in the ARDS hardware.

The full screen contains over 4000 characters. The present speed of 110 characters per second is limited by the telephone line. ARDS is capable of writing at about five times that rate.

ARDS also has vector generation capability. The screen contains 1081 by 1415 (over 1.5 million) addressable points (but not necessarily resolvable points). Setpoints position the beam at any position $-1023 \leq x, y \leq +1023$. Vectors may be drawn with $-1023 \leq dx, dy \leq +1023$. The console hardware can draw about 28 setpoints or vectors each second. There is also a short vector capability which draws vectors with $-31 \leq dx, dy \leq +31$ at about 55 per second. Vectors may be made visible or invisible, solid or dotted.

We are now using ARDS as a fast, quiet teletype. There is one major difference between ARDS and a teletype--the bottom of the page. When printed information reaches the bottom, a blob (■) signals the user that the computer is waiting for him to finish reading. When he is ready, he should hit the "ERASE" key at the upper left corner of the keyboard and the computer will proceed.

Fig. 9 Reproduction of Textual Information on Eastman Kind 1991 Silver Paper. The page format is 80 characters/line, 52 lines/page or 4,160 character/page.

A FAST CONTACTLESS OPTOELECTRONIC PRINTER

J. G. Berny, J. P. Bourgoin and A. A. Donzel
Laboratoires de Marcoussis, Compagnie Generale d'Electricite, 91, Marcoussis

ABSTRACT

A fast optoelectronic printer using the properties of holograms is described. The optical aberrations appearing in the reconstituted image are discussed. The influence of the speckle due to the laser beam used for the reconstitution of the image is studied and the quality of the reconstituted alphanumeric characters is shown. A brief description of the optical light deflector chosen is given. A prototype of such a printer is described in details.

INTRODUCTION

When one carefully examines information processing techniques, a curious feature appears immediately. A great disproportion can be seen between the processing or the transfer rate of the data and the rate of the restitution in a useful form. The data are not always needed immediately after processing. Thus, one presupposes the use of mass memories which are able to store all the numerical data during long periods of time before their transfer to terminals or peripherals. Actually the results are printed on mechanical printers whose rate is necessarily limited but which have till now satisfied the users because of the reason given above. We think, therefore, that in this field a fast printer will become very useful in the next years because of the increasing volume of the data to be processed.

The new technique of printing can include very important transforms and new concepts in that field, because of their performance. However, it is not the aim of this paper to describe new organizations of complete processing systems. We limit our ambitions to give a brief description of a new technique of printing and also a description of a prototype of optoelectronic printer.

The principle uses two main elements:

1. Electrooptic or acoustooptic light deflectors and

2. Holograms and, quite necessarily, a laser.

The following sections describe the status of the work pursued in our laboratories.

PRINCIPLE OF THE OPTOELECTRONIC PRINTER

The principle uses the properties of holographic images. It is well known that such "interferograms" are able to reconstitute complex images when illuminated by a quite coherent wavefront. More over, when the reference wavefront is rotated by an angle di, the image wavefront is also rotated with the same angle and in the same sense (Fig.1).

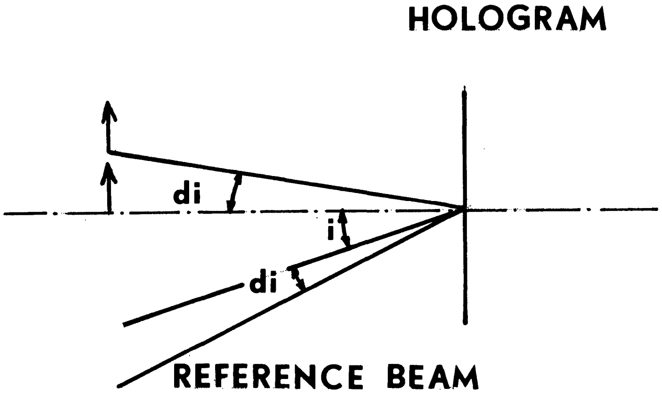


Fig. 1

Numerous means of angular light deflection have been examined in different laboratories. We only mention the main processes, because we shall speak about them a little later in more detail:

- Digital light deflectors using the properties of birefringement crystals.
- Acoustooptic cells, using Bragg diffraction in photoelastic materials.

These processes are known as "point deflectors." One can see easily that their use with holographic images convert these "point deflectors" to "image deflectors." The principle of the electrooptic printer is shown in Fig. 2. A light deflector, controlled by electronic logic, gives a choice between numerous directions of propagation of the light beams (of which two directions only are shown in Fig. 3). The coherent light beam illuminates the hologram H under two incident angles corresponding exactly to the angle of the ref-

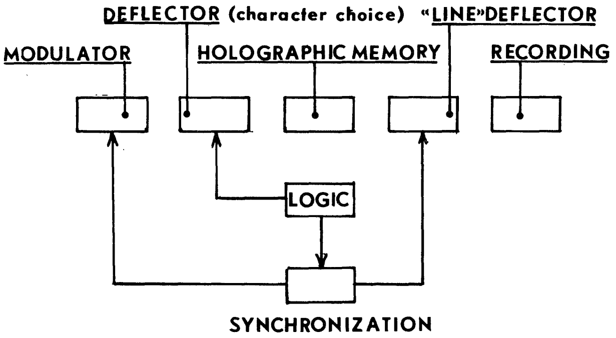


Fig.2

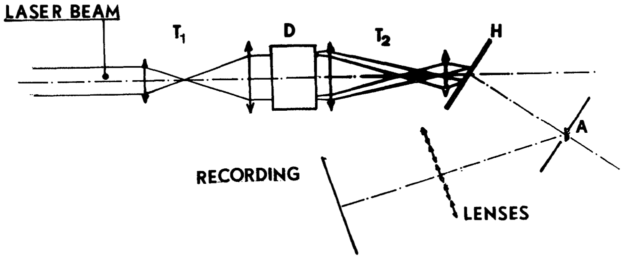


Fig.3

erence beam for each object during exposure of the hologram. The two corresponding images are reconstructed in the plane where an aperture A makes the selection of one of them. An oscillating or rotating mirror M_1 is used as "line deflector," sending the image to a point on the axis as the recording medium (for instance, photographic film) moves in the perpendicular direction to make a page of text.

THE HOLOGRAMS

We shall now study in more detail the reconstitution of the holographic images. First we will describe briefly the recording conditions of the holograms. Let us consider the interference pattern due to two coherent sources, Fig. 4. If the interfering sources are point sources, the interference pattern is made up of parallel fringes which are characteristic of their amplitudes (by means of their contrast) and of the geometry of the experiment (by means of their spatial frequency $1/i_{SO}$). The interference pattern, recorded on a photographic plate, can reconstitute the two sources. Thus a point source S (Fig. 4b) can reconstitute two images of the source O, a "direct" and a "conjugate" image. The superposition of several fringe patterns is possible. Let us call $i_{S_1O_1}$ the spatial periodicity due to coherent sources S_1 and O_1, and in the same manner, $i_{S_2O_2}$, due to the sources S_2 and O_2. But if S_1 and S_2 are placed in two different positions, O_1 and O_2 take exactly the same positions in front of the recording medium. We can see what happens when illuminating the hologram recorded in the photographic plate. Let us assume that the reconstitution is due to a source located exactly at the same location than S_1 during the recording. Then the first interference pattern ($i_{S_1O_1}$) gives the reconstitution of O_1 at its correct location and the second pattern ($i_{S_2O_2}$) gives a reconstitution of source O_2, but located in a location named O_2 in Fig. 4c. Now, let us consider a displacement of the reconstituting source S, which is put in place S_2. Thus, the first interference pattern ($i_{S_1O_1}$) gives a reconstitution of the image O located in O_1 and a reconstitution of O is located in O_2 due to the second fringe pattern ($i_{S_2O_2}$). Thus, we can say that there is a strict relation between the position S_i of the reconstituting source S in front of the interferogram (or the angle of the main beam relative to the normal of the plate) and the location O_i of the image O, relative to the interferogram. Then it is sufficient to move the reference beam SC, leaving the plate fixed in position, to obtain the choice of a determined part of the object recorded, that choice being due to a fixed aperture A, placed in the plane of the image reconstituted.

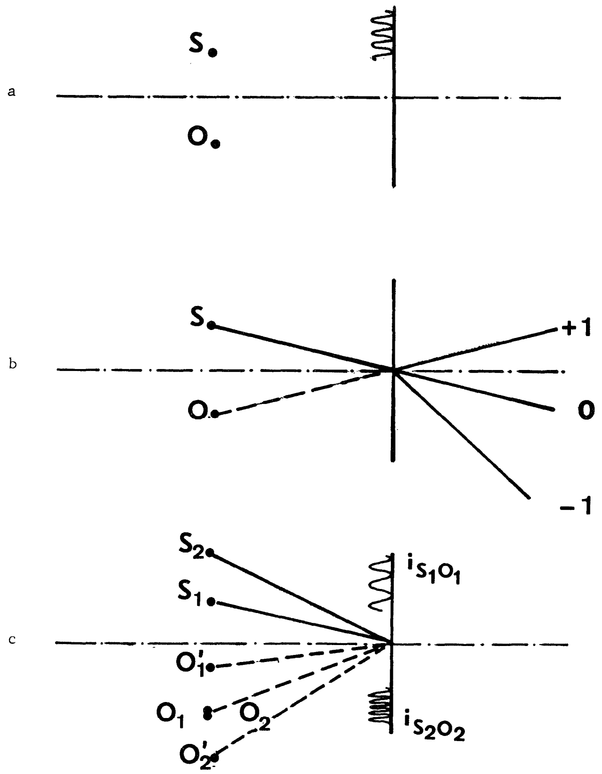


Fig 4

We can now change the point source O_i for more sophisticated objects (for instance, alphanumeric characters) and the "interferogram" becomes a hologram. The superposition of holograms is possible in two different ways. Thus, the hologram can be made in one step, all of the objects O_i being used simultaneously when recording the fringes patterns, which is known as a "coherent superposition." In another way, several holograms can be superposed in successive steps, one after the other, on the same recording plate. This is known as an "incoherent superposition." A third way to record the holograms is to make juxtaposed holograms of each object. That solution has not been chosen for reasons which will be discussed.

Optical aberrations in the reconstituted image are different in the two cases but must be taken in account in the choice of the type of holograms used. Let us consider a hologram centered in C, made by the interfering patterns due to the reference source S and a point source O (Fig. 5). The reconstitution will be due to a source located in S and for the estimation of optical aberrations, we shall call:

$SC=R \quad CS'=R' \quad CO=x \quad CO'=x' \quad OS=y \quad O'S'=y' \quad CP=h$

Optical aberrations are due to the deviation of the real beams compared to the theoretical beam or, if preferred, by the difference between the real and theoretical optical path: $\Delta = (ON) - (SN)$

A stigmatic image would remain if the following condition is realized:

$$(ON) - (SN) = (O'N) - (S'N) \qquad (1)$$

A quite stigmatic image will be obtained if the difference relative to the condition of equation (1) is small. That difference can be calculated by the set of equations:

$$\overline{ON}^2 = x^2 + (y - h\cos\varphi)^2 + (h\sin\varphi)^2$$

$$\overline{SN}^2 = R^2 + h^2$$

Putting these quantities in equation (1), using an expansion and ordering the different terms of the aberration we have:

$$\left.\begin{aligned} k\,d\Delta = & -\frac{3}{8}h^4\left(\frac{1}{R'}-\frac{1}{R}\right)\left(\frac{1}{R'}-\frac{1}{x'}\right)\left(\frac{1}{R'}+\frac{1}{x'}\right) + \\ & \frac{h^3y'\cos\varphi}{2x}\left(\frac{1}{R'}-\frac{1}{R}\right)\left(\frac{2}{x}+\frac{1}{R'}-\frac{1}{R}\right) + \\ & \frac{h^2y'^2}{4x^2}(2+\cos 2\varphi)\left(\frac{1}{R}-\frac{1}{R'}\right) - \frac{h}{2}\cos\varphi\left(\frac{y_1'^3}{x_1'^3}-\frac{y^3}{x^3}\right) \end{aligned}\right\} (2)$$

Equation (2) assumes that the hologram can be considered as the optical pupil. It shows that holographic images have optical aberrations as spherical lenses: the distortion, which is null, is the difference between the two cases. Equation (2) shows that the process called "incoherent superposition" is better, since in all the terms of the expansion containing a term $\frac{1}{R'} - \frac{1}{R}$ the aberrations will decrease and become null when R=R', which is the case.

It is not, inversely, the case of the process called "coherent superposition": the values of R and R' are necessarily different, except when they are very great or, if preferred, the wavefront is plane. The conditions of realization of holograms prohibits the use of the advantage known as "Fourier holography," the case for which x=R. So we shall use the process of coherent superposition with a plane reference wavefront, which permits the holographic memory of the set of alphanumeric characters to be made in one step.

Two kinds of holograms can be used: bleached transmission phase holograms and metallized phase reflecting holograms. The efficiency of the two kinds is similar and is defined as the ratio between the light power W_1 used to reconstruct the direct image and the light power W_0 of the beam which illuminates the hologram:

$$R = \frac{W_1}{W_0}$$

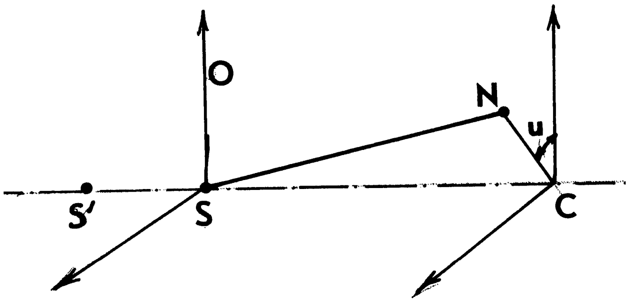


Fig. 5

Metallized phase holograms have been studied in detail because their efficiency is greater than the efficiency of amplitude holograms. The image is reconstructed[3] because of the phase distortion given to the reference wavefront reflecting on the surface of the gelatin. The surface of the plate shows a relief whose amplitude is assumed, in first approximation, to be linearly related to the optical density of the plate after chemical processing.

Therefore efficient bleaching techniques have been used and have given holograms with efficiency, defined by relation 3, around 10 percent. These holograms have two advantages, compared with metallized holograms. First, they can be placed exactly in the pupil of the optical system (which will be discussed further), and they do not show granularity due to orange peel effect, which appears sometimes with metallized holograms.

Since image quality of our metallized holograms can be considered as good, we prefer to use bleached transmission holograms. Some difficulties are due to the granularity in the image because of the diffuser needed for the exposure of the hologram. The diffuser, placed behind the alphanumeric model, illuminated by a laser beam gives in the plan of the photographic plate a speckle pattern. This pattern is responsible for the noise which appears in the image reconstructed by the laser beam illuminating the hologram. (See Reference 4 for Goldfisher's theory and a description of a process which can suppress this inconvenience.)

Taking our definition, the S/B ratio in the image can be related to the external dimensions H of the hologram and those which should give the theoretical diffraction limit H_d.

For an S/B ratio of 10, the linear dimensions of the hologram must be 10 times greater than the H_d linear dimension. Thus, the numerical aperture of the beam corresponding to the diffraction limit gives the dimension of the smallest image which could be separated y:

$$F\varepsilon = \frac{F\lambda}{H_d} = y$$

Giving to λ, y the following values

$$\lambda = 633\,\text{nm} \qquad y = 0.1\,\text{mm}$$

we find:

$$\frac{F}{H_d} = \frac{y}{\lambda} = 160$$

A mean value for F is about 100mm. Thus the smallest real dimension of the hologram side needed to obtain a ratio of 10 is around 6mm; the holograms used in the prototype will have approximatevely this dimension. One can see , fig.6, two different reconstitutions of the same hologram with two different laser beam diameter (1.0 and 6.0mm). The speckle of the photo on the left disappears in the photo on the right. The quality of images can be increased by using good quality optics which was not the case for the reconstitution shown in Fig.6, where one can see aberrations for the external letters.

1 2 3 4
5 6 7 8
9 0 + -
x : = ,
1 2 3 4
5 6 7 8
9 0 + -
x : = ,

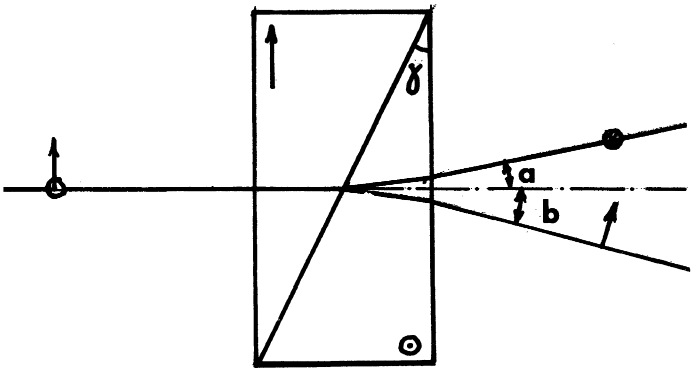
RESTITUTION WITH DIFFERENT DIAMETERS (left:1mm ; right:6mm)

Fig.6

OPTICAL DEFLECTION SYSTEM

Several processes have been examined: acoustooptic deflectors using Bragg effect in photoelastic materials and digital electrooptic deflection using birefringent crystals. The second process has been chosen for the printing prototype, because of its easier realization compared with the others. Acoustooptic deflectors are easy to make with water, but the performance is not sufficient. Thus, the search for materials having the most interesting photoelastic coefficients is necessary.
The electrooptic deflector used is composed of several cells which, when submitted to an electric voltage, turn the polarisation vector of the light beam. Each cell is followed by a Wollaston prism made with a birefringent material, which is responsible for the light beam direction in terms of the light polarization.
This processs, has been taken for our prototype. One know that a Wollaston prism is made of two birefringent prisms of same angle whose optic axis is paralell to the incident plane wavefront, (fig.7). For small incident angles one can see that the deflection angle δ is given by

$$\delta = a + b = 2\,\frac{n_e - n_o}{n}\,\mathrm{tg}\,\gamma$$



WOLLASTON PRISM

Fig.7

where n_o and n_e are the two refraction indexes of the birefringent material and n is the index of the adjacent medium.
The electrooptic cell is made of KD*P (potassium dideuterated phosphat). This crystal can be used with a longitudinal Pockels effect (electric field paralell to the optic axis and the light propagation) by use of the r_{63} coefficient of the electrooptic matrix. The cell is thus composed of a plane paralell plate with optic axis perpendicular to the face of the plate. The electric voltage needed to rotate the polarization vector of $\pi/2$ is (half wave voltage) around

$$V_{\lambda/2} = 4{,}2 \text{ kV}$$

Fig.8 shows the shematic of such a cell. The external dimensions are 15×15×2mm for the electrooptic crystal.

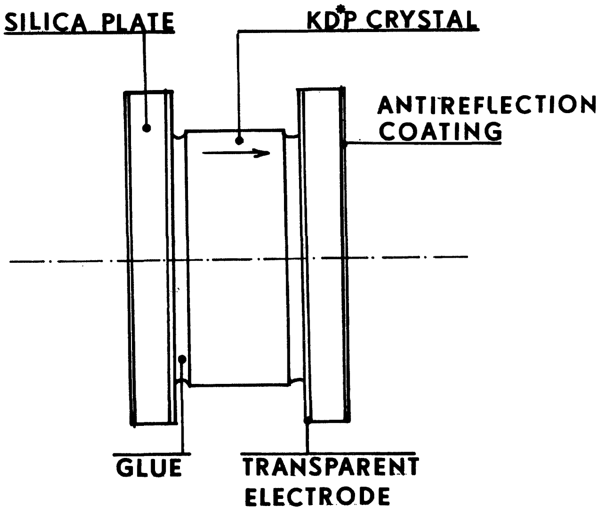


Fig.8

Transparent electrodes ,made of indium oxyde,are deposited on silica plates;two plates are pastened on each side of the electrooptic crystal.Then antireflection coatings are deposited on the free faces of the cells. The transparent electrodes have an electric square resistance of 300 ohms.The capacitance of the cell is around 70pF .The optical transmittance is 95% .
A set of 4 cells have been used to print 16 alphanumeric characters.

DESCRIPTION OF THE PROTOTYPE

As it has been said above,the principle is based on the properties of holograms and light deflectors.The holographic memory of p alphanumeric characters can be made of:

- p juxtaposed holograms of 1 character each,
- p superposed holograms.

It has been shown that the second type was preferable because of the simplification brought in the realization,but for another reason which is explained here after.It must be kept in mind that a second deflector is necessary for the inscription of a line;thus,the choice and the reconstitution of a character must not act on the position of the character in the line upon the recording film after deflection by the second deflector ,called "line deflector".
Thus,can be seen (fig.9) the importance of the pupil and also that it is preferable for the pupil to be static.Fig.9 shows a matrix of juxtaposed holograms, each of them having a proper center C_i,and the deflected beam of light illuminating one of these holograms which recontructs the alphanumeric character in the aperture centered in I.The main light ray of the beam C_iI makes with the reference axis Ix an angle b_i whose value must be taken in account,mainly to determine the correct position of the character written in a determined position on the recording film.The same dificulty appears when reflecting phase holograms are used:they must be placed necessarily at a non negligeable distance of the light deflector,thus the section of the light beam is moving on the hologram.
It is not the case for superposed holograms ,used in transmission:they can effectively be placed exactly in the exit pupil of the system.The reference beams are moving in rotation around the center C_o of the hologram This also appears in Fig.3 where a hologram is shown in the plane of exit pupil of telescope T_2.
These different reasons explains the choice of the bleached transmission superposed holograms.
Telescope T_2 is used to adapt accurately the deflection angle of the beam illuminating the hologram.Let us call $(di)_e$ the angle between two neighbouring directions emerging from the deflector and needed for the reconstitution of two neighbouring characters.Let us call also $(di)_s$ the same angle emerging from the telescope,H_e and H_s the corresponding diameters of the beam.The magnification of the telescope is g.All these quantities are related by the equations

$$(di)_s = g\,(di)_e \qquad H_s = \frac{H_e}{g}$$

The elementary deflection angles $(di)_e$ must be chosen to permit correct values of $(di)_s$ and H_s.Thus,reconstitution of different alphanumeric characters will be possible without overlapping and a sufficient image quality.
The second deflector,called "line deflector",is due to a galvanometric mirror whose frequency bandwith is very large (1. to 10 kHz).The mirror is 2x3mm linear dimensions and is used as selective aperture.The deflection of the mirror is commanded by an electric voltage synchronized with electrooptic deflector and modulator The alphanumeric character is reconstituted exactly on the plane of the mirror.The light beam,after reflection by the mirror,falls upon a set of lenses which have the same focal lenght,but giving each different image location of the character on the recording film.

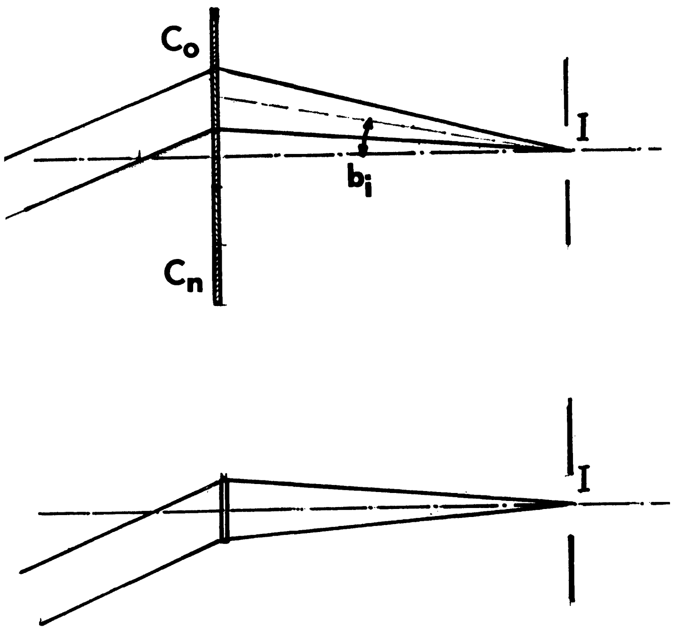


Fig.9

PERFORMANCE OF THE PRINTER

The prototype was able to print at a rate of about 200 characters/second.This rate was limited by the poor adaptation of the photographic recording with the laser wavelenght (6328 A°),and also by the rate of high voltage driving system of the electrooptic cells.
We have also done some experiments on the evaluation of the higher rate.These experiments have shown that printing rates of 10^5 to 10^6 characters/second are accessible.
The first experiment has been made in the following conditions to measure the highest possible performances A 1,5mW power HeNe laser illuminates a reflegting hologram of one character whose efficiency was around 3 %; an optical density of 22 dB was placed in the laser beam The recording photographic film was a 400 ASA Polaroid film.With these operating conditions the printing rate was 100 character/second.One can see that a magnification of the light power,the increasing of film sensitivity,the better adaptation of the source wavelenght with that of the recording film and an increased hologram efficiency leads us to think to a very fast printing system.
Thus in the same way we have made measurements of a deflection cell efficiency at high frequencies (1.5MHz)

and under sinusoidal excitation. Severe difficulties appear in these conditions, due to the glue used in the deflection cells. The H.F. heating, which appears in the glue, results in a cleaving of KD*P crystal due to a bad adaption of coefficient of dilatation between the KD*P crystal and silica plates.

CONCLUSIONS

We consider that the feasibility of a fast printer has been established. Therefore, several difficulties remain that only further experiments will help us to overcome. Three particular problems must be kept in mind:

First, the choice of the light deflector is not definitive; the appearance of new photoelastic materials ($PbMoO_4$, iodic acid) or electrooptic materials can be the origin of deflectors of higher performances

Second, the search for new recording materials could be the origin of holograms with better quality and higher efficiency.

Third, the adaptation of laser wavelength with recording film sensitivity is a very important problem; the search for new recording media is also a very important technical and economical problem.

ACKNOWLEDGEMENTS

The authors are indebted to M. Michon and R. Auffret who undertook the electrooptic study for a deflection system and to S. Chalot and G. Pitois for their very important technical assistance in the realization of the electrooptic cells.

REFERENCES

1. Leith, E. N. and J. Upatnieks. The wavefront reconstitution with diffused illumination and three dimensional objects. Jal of Opt. Soc. of America, Vol. 54, No. 11.

2. Marquet, M. and H. Royer. Les hologrammes, fascicules DRME Mai 1965.

3. Berny, J., J. Bourgoin and A. Donzel. Some considerations in photographic relief images. To be published.

4. Gerritsen, H., W. Hannan and E. Ramberg. Elimination of speckle noise in holograms with redundancy. Appl. Optics, Vol. 7, p. 2301.

5. Kulcke et al. Convergent beam digital light deflector, in Optical and Electrooptical Information Processing, Tippett et al, Ed. MIT Technology Press, Cambridge, Mass.

A HYBRID SPECIAL PURPOSE COMPUTER FOR HIGH-SPEED FFT

W. Giloi, Technical University of Berlin

ABSTRACT

The paper describes a special-purpose processor for the "fast fourier transform" algorithm (FFT) which can be used as an autonomous system as well as a special attachment to general purpose digital computers. The speciality of the processor is the hybrid implementation of the required complex arithmetic unit which provides satisfactory accuracy and a high speed of operation at minimum costs. Therefore, all data are transformed in floating point mode. All exponent operations are executed digitally, while the mantissa operations are performed by hybrid components. The high computation speed makes in many important cases on-line processing of data possible.

Spectral analysis is of vital importance in most fields of digital signal processing. In the evaluation of medical data as for example encephalograms, in speech analysis, and many other applications, thousands of data points of the frequency spectrum or estimates of the power spectral density have to be calculated for every second of recorded signals. Even when the "Fast Fourier Transform" (FFT) algorithm is used, real-time processing is far beyond the possibilities of a fast digital computer. Therefore, attempts have been made to design special purpose processors which perform the FFT-algorithm (and only that) at a much higher speed[1].

Such a special purpose FFT-processor which operates in connection with a general purpose digital computer, has four major components:

- a 'record memory' that holds the sampled records at the beginning and the computed results at the end of a calculation,
- a 'table memory' that holds the weighting factors (the trigonometric coefficients of the discrete Fourier transform),
- an indexing unit that generates the addresses of the 'record memory' and the 'table memory' in the right order, and
- a complex arithmetic unit that performs a complete FFT iteration.

The paper deals with a special purpose FFT-computer which differs from other designs published so far in the literature in two essential ways:

(I) The FFT-computer is autonomous, i.e., it is not only a special purpose hardware expansion of a general purpose computer. Thus, in addition to the above mentioned four major components, a hardware implementation of the microprograms that perform the entire FFT procedure on a given number of data points is required.

(II) The arithmetic unit which executes many times the FFT iteration (sometimes called the 'FFT butterfly') is not purely digitally implemented but in a hybrid way and, hence, provides a higher speed of operation at much smaller costs than a pure digital implementation could provide.

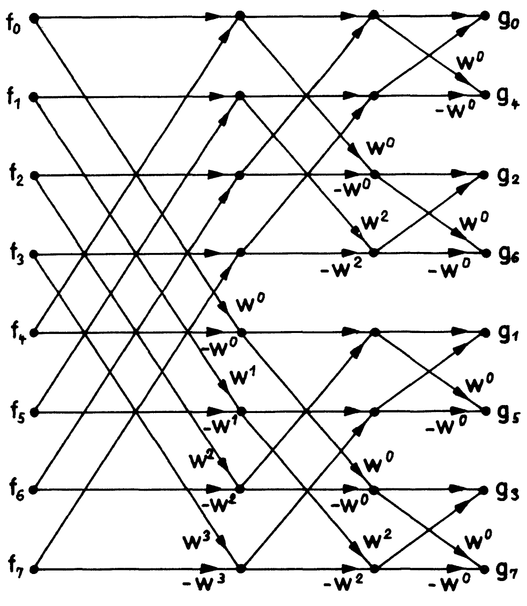


Fig. 1 Graph representation of a FFT procedure for N = 8 record samples

Fig. 1 gives a graph representation of the FFT procedure. For sake of simplicity, we assume that only $N = 8$ data point have to be transformed. Thus, $\log_2 8 = 3$ iteration steps have to be performed, and the weighting factors are in this case

$$W^n = (\exp \frac{2\pi i}{8})^n \;, \quad i = \sqrt{-1} \;, \quad n = o,1,\ldots,7.$$

In the graph representation each node corresponds with a complex addition, leading to an intermediate variable or to the final result (at the end of the last step). Each transmission represents a complex multiplication either by the complex weighting factor W^n or by 1 (the arrows of the graph are labelled by the respective weighting factors). Each iteration which transforms a pair of complex variables into a new pair of complex variables is called 'FFT butterfly' because of the pattern of its flow graph.

Each FFT butterfly can be split into 2 identical operations (except a change of the weighting factor) as it can be seen from fig. 2. If the first weighting factor is W^n, the second one is always $W^{N/2+n}$ for any value n. With $W^{N/2} = \exp(\pi i) = -1$ we have $W^{N/2+n} = -W^n$, and hence the operations of the FFT butterfly are

$$g_j = f_j + f_k \cdot W^n$$

$$g_k = f_j - f_k \cdot W^n$$

Thus, one FFT butterfly requires only one complex multiplication and two complex additions, or in terms of real operations: 4 real multiplications and 4 real additions/subtractions. If the additions or subtractions are performed on variables that are given in floating point mode, however, the necessary matching of exponents of the terms to be added and subtracted has to be done only once.

In the course of the entire FFT procedure, the iteration process executes first of all the sequence of N/2 'butterflys' of the first step (the first level of iteration), then the N/2 'butterflys' of the second step and so on. This gives the advantage of using the record memory that holds at the beginning the N record samples to be transformed as a buffer memory for all intermediate results of the first iteration step, each time overwriting the initial pair of data. That procedure is repeated at each level of iteration, leading after $\log_2 N$ steps to the final results of the transformation. As it can be seen from fig. 1, the results are not stored in their natural order, i.e. they are not corresponding with the order of addresses of the respective memory cells. But as the sequence of addresses which is now required for reading out the results in the right order equals the sequence of addresses that is required for reading the weighting factors in the right order (as the example of fig. 1 shows), one single indexing unit which generates such a sequence of addresses can be used in both cases[1].

Fig. 3 gives a general scheme of organization of the special purpose FFT-computer. The record samples and the calculated results can be arbitrarily put in or put out in analog as well as in digital form. Hence, the FFT-computer can be used as an external special purpose processor in connection with a general purpose computer, or it can be used as an autonomous device to which analog signals are put in and which gives out the Fourier transform of these signal in analog form to a tape or a plotter or in digital form to a general purpose computer for further processing, etc. The sampling of input signals is controled by a real-time clock.

If the record samples have to be put in digital form (e.g., from the memory of a general purpose digital computer), they have to be converted into the appropriate floating point mode. If they are put in in analog form, they are first of all automatically scaled by an automatic range-selection ADC and then converted into digital form by an analog-to-digital converter ('subrange ADC'). The digital output of the range selection ADC gives the exponent while the 'subrange ADC' output gives the mantissa. As the original record is assumed to be analog in nature, a 16-bit representation is considered as being more than satisfactory (leading to an error of $2^{-16} \approx .0015$ %). Thus, the word length of the record memory is 16 bits, and these 16 bits are split into an 11-bit-mantissa (including sign) and a 5 bit exponent. The exponent is by definition always negative, and hence, its sign does not have to be carried.

If the calculated results have to be put in digital form, they are given in the above mentioned floating point representation. If the output is required in analog form, the results are reconverted into fixed point mode using one of the DACs (which are required in the arithmetic unit anyway) in connection with the range-selection device which works now as a range-control amplifier.

A general error analysis of the FFT algorithms has shown[2] that in order to avoid error accumulation caused by the intrinsic iteration process, the accuracy of the arithmetic operations (especially of the multiplications) and of the trigonometric weighting factors should be higher than the accuracy of the record samples. Therefore, the table memory that holds the trigonometric coefficients has a word length of 18 bit (13-bit-mantissa + 5-bit-exponent), and the accuracy of the MDACs is better than these 13 bits (analog error < .005 % full scale). The table memory is a roped read-only memory the capacity of which is 4 K words, and the cycle time is about 2.5 μs. As the table memory output is buffered, the memory cycle time is of no significance, since the next value is fetched while the execution of a FFT butterfly is still going on.

All calculations are performed in floating point mode in order to obtain satisfactory accuracy and avoid the problems of scaling and overflow (if the arithmetic operations would be solely performed by analog components, we had to face both problems). The hybrid implementation of the complex arithmetic unit garantees satisfactory accuracy and a very high speed of operation at minimal costs: while all mantissa multiplications are performed in parallel by multiplying digital-to-analog converters (MDACs), the exponents are added or subtracted by digital adders.

The addition and subtraction of floating point numbers according to the equations in fig. 2, however, are performed by analog summing amplifiers. Such a procedure requires, of course, a matching of the exponents of the two numbers to be added or subtracted. In our FFT computer, this is done by building the difference of the exponents of both terms (the 5-bit-adder networks are, therefore, also used for subtraction and comparison of two numbers). The difference is used in order to rescale the mantissa of the smaller of both terms so that it corresponds with the exponent of the greater variable, and thereafter, addition or subtraction of the mantissa of both terms can be performed. The scaling of the smaller variable is performed by a 5-bit-MDAC.

Fig. 4 gives a block diagram of the hybrid complex arithmetic unit. The unit executes sequentially all required operations of one FFT-butterfly in the following steps:

step 1 Ma_k and Mb_k are loaded into the 11 bit-DACs (while Mc_n and Md_n are already located in the input registers of the MDACs). Simultaneously, Eb_k, Eb_n and Ea_k, Ed_n are loaded into both ADD/COMPARE units. At the end of step 1 we have Ma_kc_n, Mb_kc_n, Mb_kd_n, and Ma_kd_n in analog form and Eb_kd_n, Ea_kd_n in digital form (stored in a buffer register).

step 2 Ea_k, Ec_n and Eb_k, Ec_n are loaded into the ADD/COMPARE unit and Ea_kc_n, Eb_kc_n are calculated.

step 3 Ea_j and Eb_j are loaded into the ADD/COMPARE units and the 3 components Ea_j, Ea_kd_n, Eb_kd_n and Eb_j, Eb_kc_n, Ea_kd_n, respectively, are compared. The result controles the scalers so that the smaller two of all three components are matched to the greatest one. The mantissa additions and subtractions are performed leading to the results g_j and g_k according to eq. (3) in fig. 2.

The ADCs, MDACs, and scaling amplifiers (which are actually MDACs too) are not only very accurate but also extremely fast. The digital settling time of the MDAC to full accuracy (14 bit or .005 %) is less than 1 μs; the 3-dB-frequency of the analog path is 5 MHz! The ADC which re-converts the calculated results into digital form requires approximately a conversion time of 1,5μs. These components have been developed in the author's laboratory under 'project PHENO'; their principles of operation and the obtained results have already been published[3].

REFERENCES

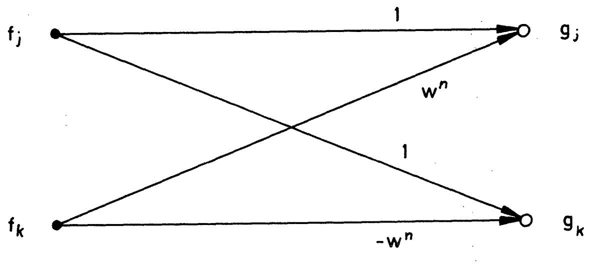
1 KLAHN, R., SHIVELY, R.R., GOMEZ, E., GILMARTIN, M.J., "The Time-saver: FFT Hardware", Electronics, June 24, 1968, pp. 92-97

2 RESCH, G., "FFT Programm in Metasymbol", Diploma Thesis 1969, Institute of Information Processing, Technical University of Berlin

3 GILOI, W., SOMMER, H., "PHENO- A New Concept of Hybrid Computing Elements", Proc. FJCC 1967, pp. 23-32

4 WELCH, P., "A Fixed-point Fast Fourier Transform Error Analysis", IEEE Trans. on Audio and Electroacoustics, Vol. AU-17, No. 2, June 1969, pp. 151-157

5 WEINSTEIN, C., "Roundoff Noise in Floating Point Fast Fourier Transform Computation," IEEE Trans. on Audio and Electroacoustics, Vol. AU 17, No. 3, Sept. 1969.



Komplex Notation of a FFT-Butterfly:

$$g_j = f_j + f_k \cdot W^n$$

$$g_k = f_j - f_k \cdot W^n$$

Real Notation of a FFT-Butterfly:

$$f_k = a_k + ib_k, \qquad W^n = c_n + id_n$$

$$f_k \cdot W^n = \{a_k c_n - b_k d_n\} + i\,\{b_k c_n + a_k d_n\}$$

$$g_j = \{a_j + a_k c_n - b_k d_n\} + i\,\{b_j + b_k c_n + a_k d_n\}$$

$$g_k = \{a_j - (a_k c_n - b_k d_n)\} + i\,\{b_j - (b_k c_n + a_k d_n)\}$$

Floating Point Notation:

M denotes the mantissa; e.g. Mf_k = mantissa of f_k

E denotes the exponent; e.g. Ef_k = exponent of f_k

$a_k = (Ma_k, Ea_k)$; $\qquad b_k = (Mb_k, Eb_k)$;

$c_k = (Mc_k, Ec_k)$; $\qquad d_k = (Md_k, Ed_k)$;

$f_k = (Mf_k, Ef_k)$; $\qquad g_k = (Mg_k, Eg_k)$.

Figure 2 The operations of one FFT "butterfly" (one iteration)

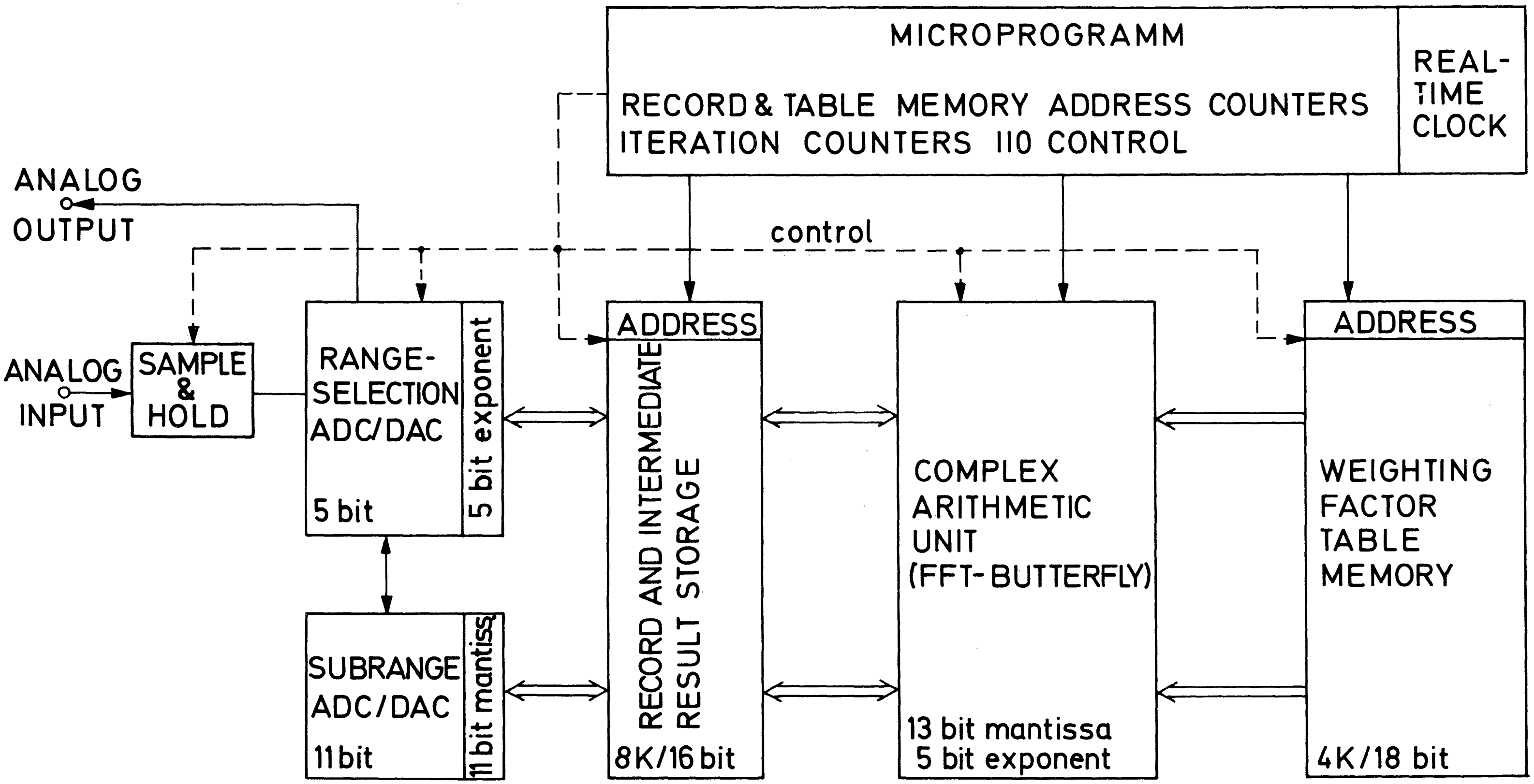

MICROPROGRAMM
RECORD & TABLE MEMORY ADDRESS COUNTERS
ITERATION COUNTERS I/O CONTROL
REAL-TIME CLOCK
ANALOG OUTPUT
control
ANALOG INPUT
SAMPLE & HOLD
RANGE-SELECTION ADC/DAC
5 bit
5 bit exponent
SUBRANGE ADC/DAC
11 bit
11 bit mantissa
ADDRESS
RECORD AND INTERMEDIATE RESULT STORAGE
8K/16 bit
COMPLEX ARITHMETIC UNIT (FFT-BUTTERFLY)
13 bit mantissa
5 bit exponent
ADDRESS
WEIGHTING FACTOR TABLE MEMORY
4K/18 bit

Figure 3 General organization scheme of the special purpose FFT computer

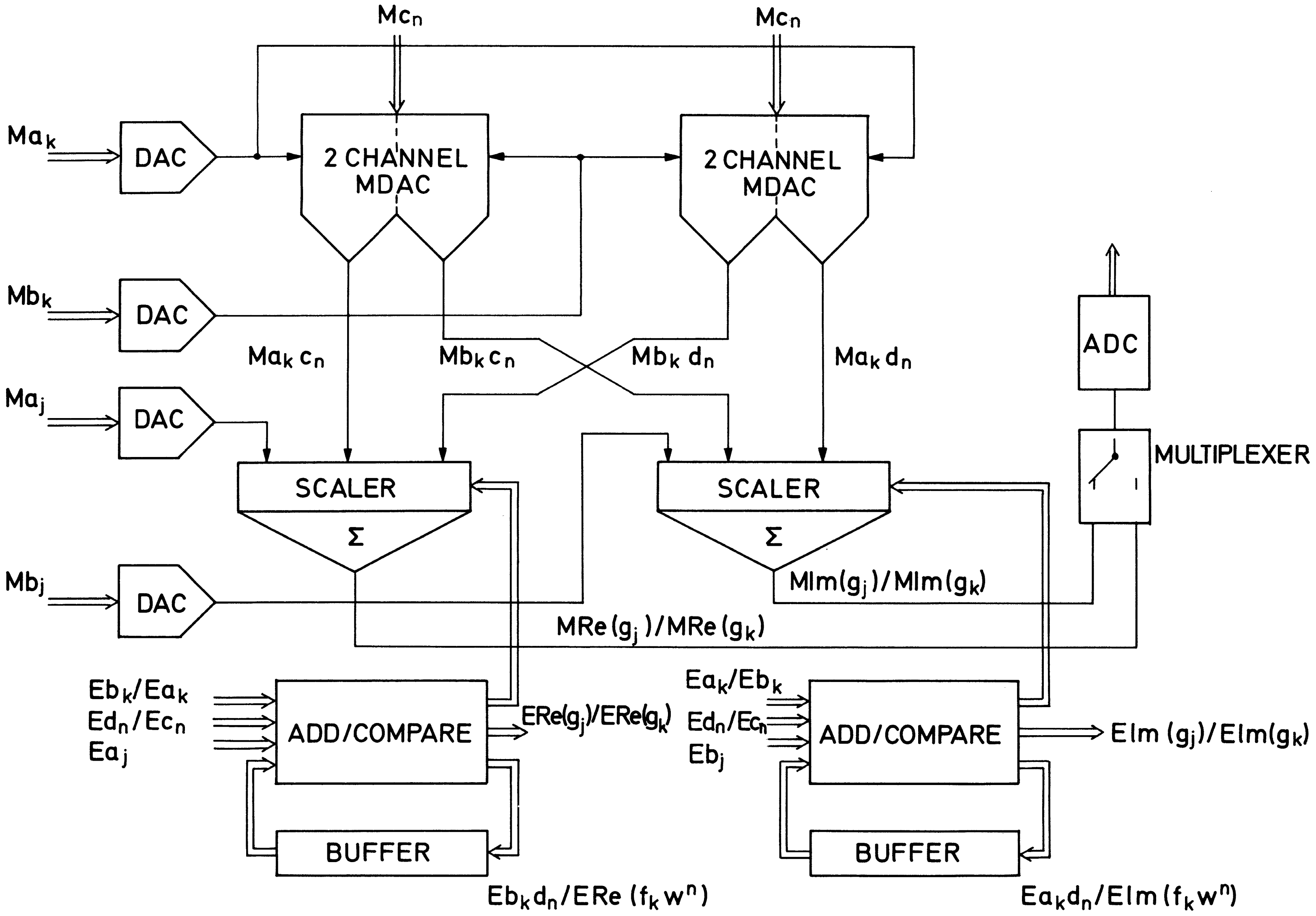

Mc$_n$
Mc$_n$
Ma$_k$
DAC
2 CHANNEL MDAC
2 CHANNEL MDAC
Mb$_k$
DAC
Ma$_k$ c$_n$
Mb$_k$ c$_n$
Mb$_k$ d$_n$
Ma$_k$ d$_n$
ADC
Ma$_j$
DAC
MULTIPLEXER
SCALER
SCALER
Σ
Σ
Mb$_j$
DAC
MIm(g$_j$)/MIm(g$_k$)
MRe (g$_j$)/MRe (g$_k$)
Eb$_k$/Ea$_k$
Ed$_n$/Ec$_n$
Ea$_j$
ADD/COMPARE
ERe(g$_j$)/ERe(g$_k$)
Ea$_k$/Eb$_k$
Ed$_n$/Ec$_n$
Eb$_j$
ADD/COMPARE
EIm (g$_j$)/EIm(g$_k$)
BUFFER
BUFFER
Eb$_k$d$_n$/ERe (f$_k$ w^n)
Ea$_k$d$_n$/EIm (f$_k$w^n)

Figure 4 Hybrid realization of the complex arithmetic unit

MONOSCOPE CHARACTER GENERATION

J. H. Selander,
Communications and Data Processing Operation, Raytheon Company

ABSTRACT

The monoscope is a cathode ray device which is used to generate alphanumeric characters and special symbols for cathode ray tube (CRT) display terminals or for hard copy printout. The monoscope is similar to a direct view CRT except that its physical size is smaller, and the phosphor screen is replaced by an alphanumeric target plate containing the character repertoire. This target serves as a very high resolution character, read only memory. The distinct advantage of the monoscope character generator is its ability to produce large size, well formed, and natural looking characters or special complex symbols less expensively than other techniques. This advantage is particularly useful in low cost alphanumeric display terminal applications where legibility and high operator accuracy of identification or script writing are required. A description of monoscope character generator techniques and design considerations follows.

INTRODUCTION

The monoscope is a special kind of read only memory; when it is used for character generation, it is capable of providing very high resolution and character legibility. Complex symbols or script writing may also be produced by selecting the appropriate target font. Emerging MOS and LSI technology employing dot or stroke methods cannot economically duplicate the legibility and flexibility of the monoscope character font at this time. The purpose of this article is to describe the operation and design considerations for monoscope character generation.

OPERATION

The monoscope is a cathode ray device that can be used to generate the video output signals for character generation. Two types of monoscopes[1] are generally used: the first type employs a stencil which uses the current of the electron beam to provide the video signal; in the second type, the output signal is generated by a difference in secondary emission between the target background and printed characters as shown in Figure 1, Secondary Emission Monoscope. The latter type is more commonly used for character generation in display terminals. As the electron beam lands on the target surface, a difference in secondary emission of electrons from the character and the background provides a video signal. The secondary electrons emitted are attracted to the collector which is at a slightly more positive voltage. A lower current flow occurs when the beam hits the printed character portion.[2] As the electron beam scans across the character for a given scanning pattern, the entire character is repeated on a direct view cathode ray tube. The monoscope target typically contains an eight by eight matrix of characters, or a total of 64 character positions. Ninety-six character fonts with upper and lower case characters are also available. To obtain maximum output from a monoscope, the target plate and printed character material have secondary emission ratios differing as widely as possible.[3] Variation of the monoscope high voltage will vary the secondary emission ratios as shown in Figure 2, Variation of Secondary Emission Coefficient with High Voltage. The top curve represents the target aluminum base material, and the bottom curve represents the printed carbon material at the characters. The dotted line indicates a typical high voltage value used, optimizing secondary emission differences, electron beam resolution, and deflection sensitivity.

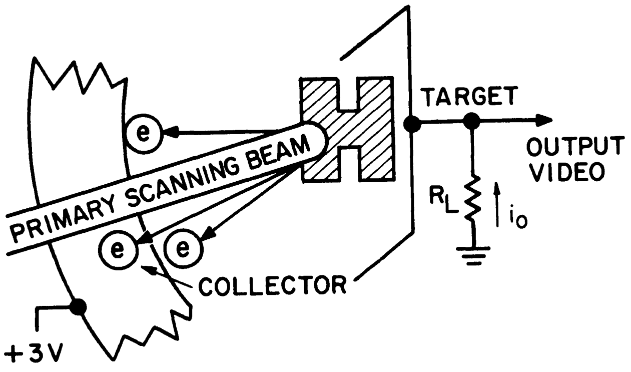


Figure 1. Secondary Emission Monoscope

System Configuration

The monoscope character generator scan can be synchronized to the CRT display scan by at least two methods. A standard television type raster may be employed where the monoscope characters are selected and scanned at a rate

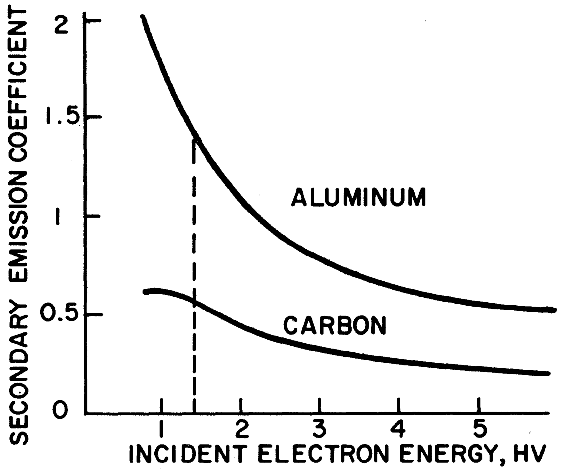


Figure 2. Variation of Secondary Emission Coefficient with High Voltage

equivalent to the television monitor synchronization. In this case, the characters are formed one character line at a time, with several lines required to complete a row of characters.

A second method[4, 5] uses a raster scan with a high frequency vertical diddle scan impressed upon it as shown in Figure 3, Monoscope Character Generation Technique. The contents of a memory are accessed and applied to X and Y digital to analog converters (three bits each axis) under control of the CRT scan pattern timing. A character will be selected according to the code of the six bits removed from memory (see Figure 4, Monoscope Character Generator). The video signal obtained from the monoscope target is amplified and used to unblank the CRT. The diddle scan is fed simultaneously to both the monoscope and the CRT vertical deflection systems. On some display terminals, the diddle character scan is generated on the CRT by a separate deflection system. If the number of displayed characters is low (less than 500), then the diddle scan can be mixed with the main CRT deflection system since bandwidth requirements are not too stringent. With this raster method, the horizontal and vertical timing requirements are such that the horizontal raster lines on the CRT display will tend to slope down as shown in Figure 5, CRT Display Raster Skew. To correct this, the horizontal signal is fed into the CRT display Y deflection amplifier, providing a straight horizontal raster line.

DESIGN CONSIDERATIONS

All character generators for CRT display terminals require a conversion of digital data (containing the character selection codes for each character position on the CRT screen) to the analog signals required by the CRT display. Since the monoscope is an analog device, the digital to analog conversion occurs at the input of the character generator.

Positional Stability

The typical monoscope contains an eight by eight matrix of characters. With this matrix arrangement, eight discrete steps for X and Y are required. The high frequency diddle scan is summed with the digital to analog converter output at the monoscope Y deflection amplifier. To maintain positional stability of the character selection and diddle character scan, the monoscope accelerating voltage should be regulated and stable digital to analog converters and deflection amplifiers employed. These characteristics can be readily achieved by use of feedback techniques.

Video Output Stability

The monoscope beam current is usually limited to one or two microamperes. This current should be held relatively constant by use of a cathode degeneration resistor. Since the target

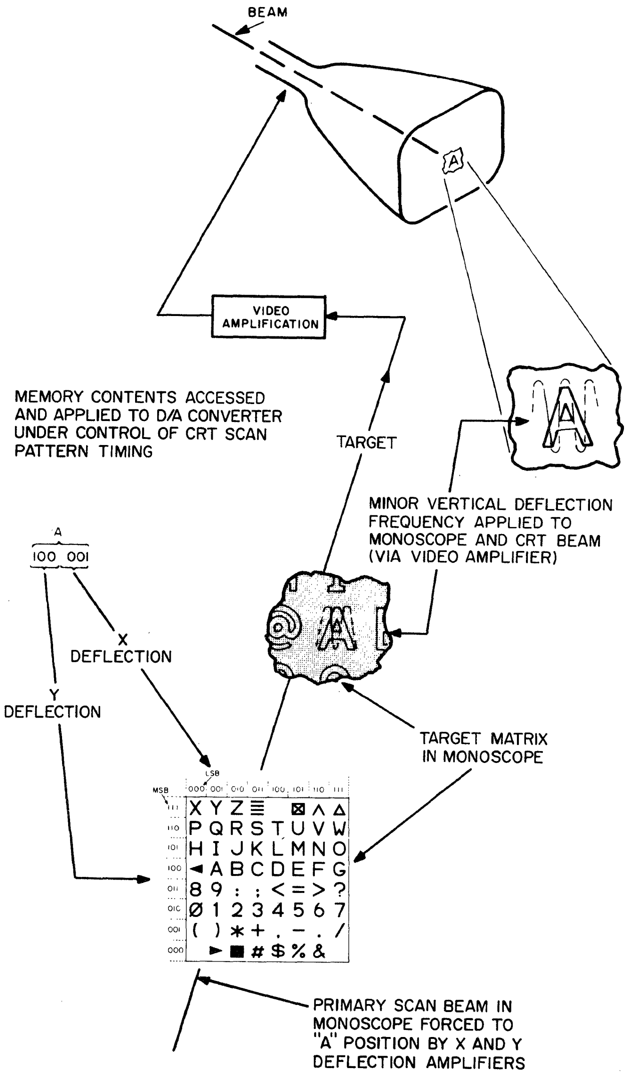

BEAM
VIDEO AMPLIFICATION
MEMORY CONTENTS ACCESSED AND APPLIED TO D/A CONVERTER UNDER CONTROL OF CRT SCAN PATTERN TIMING
TARGET
MINOR VERTICAL DEFLECTION FREQUENCY APPLIED TO MONOSCOPE AND CRT BEAM (VIA VIDEO AMPLIFIER)
A
100 001
X DEFLECTION
Y DEFLECTION
TARGET MATRIX IN MONOSCOPE
LSB
MSB
000 001 010 011 100 101 110 111
111 X Y Z ≡ ⊠ ∧ △
110 P Q R S T U V W
101 H I J K L M N O
100 ◄ A B C D E F G
011 8 9 : ; < = > ?
010 0 1 2 3 4 5 6 7
001 ( ) * + . − . /
000 ► ■ # $ % &
PRIMARY SCAN BEAM IN MONOSCOPE FORCED TO "A" POSITION BY X AND Y DEFLECTION AMPLIFIERS

Figure 3. Monoscope Character Generation Technique

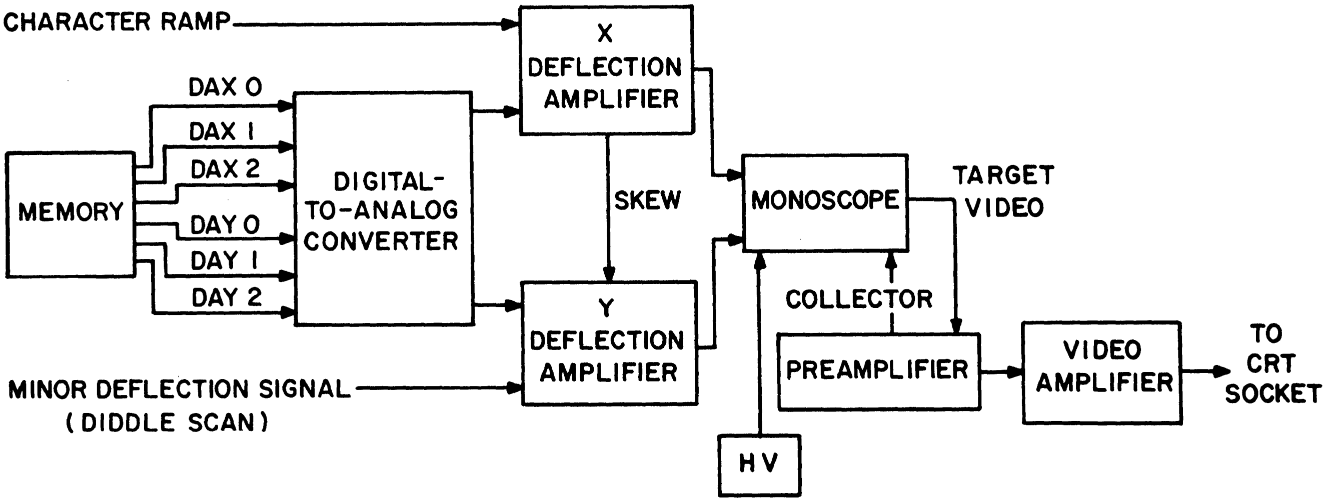

CHARACTER RAMP
X DEFLECTION AMPLIFIER
DAX 0
DAX 1
DAX 2
DAY 0
DAY 1
DAY 2
MEMORY
DIGITAL-TO-ANALOG CONVERTER
SKEW
MONOSCOPE
TARGET VIDEO
COLLECTOR
Y DEFLECTION AMPLIFIER
PREAMPLIFIER
VIDEO AMPLIFIER
TO CRT SOCKET
MINOR DEFLECTION SIGNAL (DIDDLE SCAN)
HV

Figure 4. Monoscope Character Generator

signal from the monoscope is relatively low (two to three millivolts with target R_L of 1000 ohms), the monoscope preamplifier should be physically located next to the monoscope target terminal. The monoscope and preamplifier are both shielded to respectively prevent erroneous deflection of the beam from stray magnetic fields and to eliminate spurious noise pickup. The preamplifier as well as the final CRT video amplifier should employ feedback to obtain video output stability.

Target Life

The secondary emission characteristics of the monoscope target tend to degrade with repeated exposure to high beam currents. For this reason the beam current is limited, typically to approximately one microampere. This will provide a target current of two to three microamperes, which is consistent with the noise level of the input stage of the preamplifier. To further limit degradation of the target, continuous sweeping of one character during blank character positions on the CRT display should be avoided. Deflecting off the active target screen area or continuous shifting to another character during each blank character on the CRT display can provide several years of normal operating life.

Target Printing

Variations in the output video signal can occur when scanning across the target. This characteristic, which is called shading, is greater on materials with higher secondary emission coefficients. For the two materials shown in Figure 2, Variation of Secondary Emission Coefficient with High Voltage, the carbon material having a lower secondary emission coefficient should be used for the characters and the aluminum material used for the background. This will provide a more uniform output signal level.

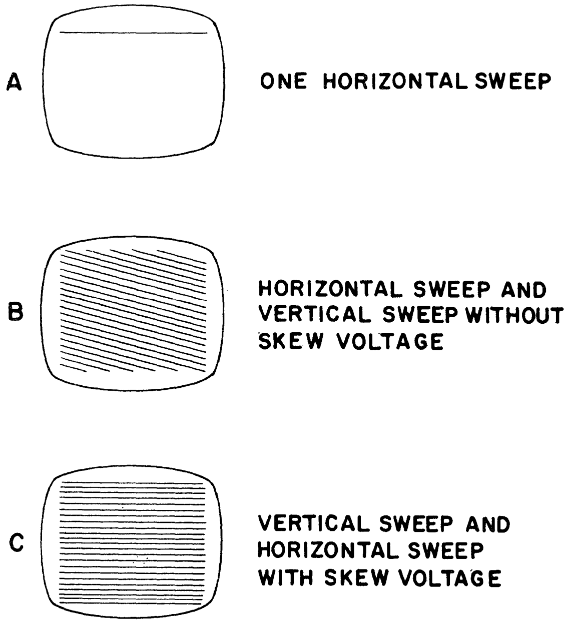

A
ONE HORIZONTAL SWEEP
B
HORIZONTAL SWEEP AND VERTICAL SWEEP WITHOUT SKEW VOLTAGE
C
VERTICAL SWEEP AND HORIZONTAL SWEEP WITH SKEW VOLTAGE

Figure 5. CRT Display Raster Skew

Video Signal

The output video signal from the monoscope target can be amplified and used directly to unblank the CRT display. However, some video processing of the raw video signal is recommended to assure constant brightness for one character and from character to character over extended periods of time. Because of the high gain of the monoscope preamplifier and video amplifier combination, ac coupling is usually utilized. The video signal, therefore, should be dc restored, and the restoration level can be referenced to a particular area of the monoscope target such as the side of the target font or the next character. The restoration time should be less than the intercharacter time. The base line can then be clipped and the peak unblanking level limited slightly to provide a uniform output level.

Amplifier Gain

The signal level at the monoscope target is typically two microamperes, giving two millivolts across a 1000 ohm load. If the unblanking signal level for the CRT display is set at 40 volts, a voltage gain of 86 dB is required. If the monoscope target and CRT control grid are physically separated from the video amplifier, undesired capacitance loading will occur at the input and output. For these reasons, separation of the video amplifier is recommended with a preamplifier located at the monoscope target and the final video amplifier located at the CRT socket. Feedback techniques should be employed for both amplifiers. The bandwidth of the amplifiers is dependent on the character writing time, number of character scanning lines, and resolution requirements; this would typically be 15 MHz for a high resolution 500 character display and 30 to 35 MHz for 1500 display characters.

Resolution

With monoscope beam currents of one to two microamperes, a five mil spot size can be obtained at the target even at relatively low accelerating voltages. A target character matrix size of 1.25 x 1.25 inches (3.175 x 3.175 centimeters) will provide a resolution of over 60,000 points. Each character of an eight by eight matrix can then have over 500 resolvable points when 20 to 25 scanning lines are used. This points out the distinct advantage of the monoscope character generation technique. Dot matrix or stroke character generation techniques, even with equivalent resolution, cannot reproduce diagonal lines without disjointed straight lines. The technique commonly used to overcome this objection is to reduce the size of the character on the CRT display. This compromises visual acuity and degrades maximum display data capability, particularly for well formed characters or complex symbols.

CONCLUSION

The monoscope is a useful and reliable source of character and symbol data. Very high resolution and character legibility can be obtained. Although the monoscope may appear to be an anachronism in comparison with MOS and LSI technology, it has distinct advantages for reproduction of complex characters. Certain design precautions are all that is necessary to provide a useful character generator for low cost display terminals and hard copy printout.

ACKNOWLEDGMENT

The author would like to express his appreciation to Mr. Derek Chambers for many helpful suggestions and comments.

REFERENCES

1. Burnett, C.E., "The Monoscope," RCA Review, 2, 414-420 (1938).

2. Raytheon Company, Components Division, "Symbolray* Character Generating Cathode Ray Tube" (CK 1414), A Technical Information Brochure, April 15, 1966.

3. Nixon, R. D., "The Monoscope," IEE Proc., 99, Part 3A, 132-135, (1952).

4. Jones, Earl D., "Character Generator For Digital Computers," Electronics, 117-120, February 12, 1960.

5. Lorenzo, Childress S., "Selecting the Best Character Generation Technique," Electro-Optical Systems Design, 24-30, March, 1970.

*Trademark of Raytheon.

LSI REFRESH MEMORIES

Enrique Moreno, Vice President, Beehive Medical Electronics, Inc.

As the number of computer installations rapidly increase, the need for a small, highly reliable, low cost display terminal becomes a necessity. The utilization of these terminals in small computer systems will depend heavily on their cost and flexibility. LSI semiconductor memories will play a significant role in the implementation of such a computer display terminal. Because of their high performance, high reliability, small size and weight, and low cost, LSI memory devices satisfy the basic system requirements. The devices can readily be adapted to systems of varying size and organization to provide memory systems whose price per bit element remains essentially constant for any number of bits required. Interface with conventional logic families can easily be accomplished. The signal compatibility between logic and memory elements eliminate the need for especialized circuits normally required with other memory devices.

LSI memories are presently configured using bipolar construction or MOS (metal oxide silicon) construction. Bipolar memories basically are multiple-emitter transistors connected as a flip-flop. They typically exhibit extremely fast cycle times (150 ns) and access times (50-100 ns). Power dissipation typically runs 350-420 milliwatts.

MOS memory devices are available in static and dynamic configurations. In static memory devices, the data is held indefinitely without clock or strobe pulses. From the input, the array looks like a capacitor requiring a finite time to charge and thus has an inherent access time of about 3 to 4 microsecs. Static arrays dissipate approximately 150 microwatts of power per bit.

Dynamic memories require periodic restoration at some minimum clock rate. However, they typically provide twice the bit density per given area, five to ten times the speed, and one fourth the power dissipation of static memories. Dynamic memories are expected to cost at least half the price of static devices.

The modular plug-in construction of MOS memories offer another distinct advantage when used in systems requiring various memory sizes and fast availability of data. Thus, the shift register is the semiconductor equivalent for delay lines or drum memories. However, unlike their mechanical counterparts, the shift register clock can be varied in time or can be stopped.

MOS memories have definite advantages for refresh memory and READ-ONLY Memory (ROM) applications.

Read-Only memories can be economically fabricated with thousands of bits per chip. They are replacing various system functions such as code conversions, table look-up, and microprogramming. Easily configured, the ROM has become a standard for specific computer terminal memory applications.

Dynamic MOS memory devices provide an interesting approach for refresh memory in display terminals. With the information circulation through the memory, it is constantly available for use. Inexpensive TV circuitry can be utilized for the video and deflection requirements. Editing, formatting, I/O functions are all simplified by circulating data through a given point in time.

The use of LSI memories has proven to be very flexible and simple to integrate with the rest of the logic. They are reliable and economical. The LSI memory manufacturers anticipate significant reductions in price and increased complexity. As users of LSI memory devices, we agree that the application of these devices and therefore the development and utilization of associated circuitry by the terminal and other peripheral manufacturers that will expedite the realization of very economic and reliable LSI memories.

GRAPHIC DISPLAYS USING IMAGE STORAGE TECHNIQUES

Thomas B. Cheek - Computer Displays, Incorporated

Ever since computers were first developed the problem has existed of how people should communicate with these complex machines. For a long while, computers and computer time were so expensive that almost all the effort devoted to bridging the communication gap was directed to improving machine efficiency.

The situation is now rapidly changing. Hardware costs are declining steadily while manpower becomes increasingly expensive. Designers can now improve utilization of human resources by building equipment aimed at making it easy for users to state problems and to get answers.

The major system innovation in recent years that has helped narrow the communications gap is the advent of time-sharing systems. As soon as it was demonstrated that time-sharing systems could provide fast and simple man-machine interaction, attention was focused on the terminal.

The terminal in any time-sharing system is the user's window to the computer and in large measure the capability of the terminal limits the capability of the system. Certainly this is true of the noisy, 10 character-per-second teletype that is still the most common type of time-sharing terminal.

What is needed is a new "universal" terminal that can be mass produced at low cost and can be used in many different systems to do many different jobs. A number of characteristics of such a terminal are generally agreed upon: the terminal should be noiseless, should present data quickly, should be usable in remote locations over phone lines, and should be inexpensive enough to be used in large numbers. Other features deemed desirable by many are that the standard information code used between the terminal and computer be ASCII and that the symbol set include all 96 printable symbols of the ASCII set. Finally, there are many who believe that graphics - the ability to draw high quality pictures - is an extremely valuable capability in a general-purpose terminal that might be used in any scientific, engineering, educational, financial, or management information system.

The various attempts to produce an advanced terminal have generally required some compromise in attaining the above list of "ideal" characteristics. Usually the "low cost" requirement forces other features to be dropped. Thus, there are many alphanumeric-only terminals because graphics is expensive. Another technique used to lower cost is to cluster a number of displays around a shared controller or mini-computer, saving money through shared hardware. The problem here is that users often want isolated terminals and very few time-sharing systems are designed to interact with "cluster terminals".

Traditionally, the cost problem with CRT graphic displays has stemmed from the need for fast deflection systems and large memories to refresh the image on the screen frequently enough to avoid flicker. Within the last two years, however, several terminals have been introduced that use special "storage" CRT's that display the picture drawn on the screen without refreshing, thereby requiring only slow (and low cost) deflection amplifiers and no extra memory. These terminals sell for under $10,000.

Even at $10,000 (about 10 times the cost of a teletype) these new terminals can be cost-effective because the total terminal expense is only 10% to 30% of the cost of a large computer system. The new terminals increase throughput considerably. In particular, graphics can enormously speed up operations which involve interpretation of data -- such as circuit design and structural analysis -- by plotting graphs or line drawings instead of presenting tabular data only. Even in operations normally considered to be "text-only", graphics can be quite helpful by extending standard symbol sets to include plotted symbols such as Greek and Cyrillic letters and mathematical and chemical symbols.

The image-storage technique has also proven popular because the user can plot large and complex pictures; the only limits to data storage and display are the size and resolution of the display screen. For example, text blocks comparable to typewritten sheets (4000 symbols) are easily plotted and the presentation is steady and flicker-free. Intricate pictures with thousands of line segments -- such as integrated circuit masks -- can be handled equally well.

The image-storage graphic terminals are well suited at present to general time-sharing use but a number of improvements are nevertheless needed. More compact terminal designs, but with brighter and larger screens, are desirable. Selective erasure would be helpful in many applications. System problems also exist: many time-sharing systems cannot support the higher data rates these terminals demand to be most effective. Also efficient graphics-oriented software is a necessity if users are to get full benefit from such terminals without having to become programming experts.

CENTRALIZED STORE REFRESH MEMORIES ... A PROVINCIAL VIEW

N. Clark TEC, Incorporated

ABSTRACT

The cost of memory in low cost teledisplays requiring refresh is a significant factor in total cost, but not enough to outweigh other considerations including the lower cost of centralized refresh memories. The reasons for this situation are discussed together with some applications background and a prediction.

ON CENTRALIZED REFRESH MEMORIES

A very important factor in the cost of low cost teledisplays is the refresh memory required for most cathode ray tubes. Centralized store refresh memories are less expensive than individual memories for each display, yet popularity wanes for the centralized version. Reasons are that factors other than price become involved in the choice of refresh memories not the least of which is the changing technology and systems considerations. In this context of memory use in low cost teledisplays, surely nearly everyone knows a teledisplay is used with a computer; and it's no good without software; that ASCII exists; that there are page formats; message formats; character formats; keyboard arrangements; edit functions; dot matrix characters; stroke characters; error recovery; hard copiers; expansion modules; and an operators chair. Low cost is clearly between 1300 and 7000 USA dollars because that includes all TEC products and more than enough others.

Centralized store refresh memories are, however, quite another thing. Although low in cost, such memories are less popular because of remote terminal pricing considerations. And, pricing is important! Judging from the Madison Avenue marketing of low cost teledisplays as computer terminals, it is probably _the_ most important factor. The serious buyer must contend with:

a. _Astounding claims_ (who knows what's meant by: IBM compatible; more reliable; better looking; universal interface?)

b. _Weird pricing_ (your choice: display stations at $4125 each plus a controller at $36,000 or; $50 per month for a microprocessor, communications adaptor, video subsystems, video display, and keyboard!)

c. _Strange variety_ (automatic multiple output, modular construction expansion unit Model 3, conversation mode, ASCII-8, high-contrast, ad infinitum)

After accounting for all the astounding claims, weird pricing, and strange variety of teledisplays, the serious buyer is still left with the most important question. What's it cost? The serious buyers must be amused at the agility exhibited by the engineering and marketing people of our industry on a "hot tin roof" fired by ... LOW PRICE. For this reason, we manufacturers refer to serious buyers as S. B.'s.

Unfortunately S. B.'s are scarce in my province. Are they trapped by the 200 other manufacturers of low cost teledisplays?, or with all the astounding claims, weird pricing, and strange variety, are they simply confused? ... or snowed? For those S. B.'s who can't wait, it's easier to just call your friendly TEC representative.

He will tell you that one of the biggest items of cost in a low cost teledisplay is memory. Get the memory cost down, and the teledisplay cost goes down.

Now, very inexpensive memories can be included with the viewing screen itself, e.g. storage tubes, glow discharge arrays, liquid crystal tubes, etc. A major problem with these devices is provision for electrically reading the memory. And without considering the question: who needs to?, we dismiss the whole category. The only kind of products TEC has uses cathode ray tubes requiring frequent refresh. Therefore, the memory required must completely dump at the refresh frequency which is generally 60 times per second. For a 512 character memory, the dump rate must exceed 30,720 characters per second or one character every 33 microseconds. Memories capable of this are magnetic core, mechanical delay lines, MOS shift registers, and magnetic disks.

Each non-storage type CRT display, with unique data, must have such a refresh memory. Many engineers working for many years on the least expensive way to do this have also had to consider such things as:

a. space and mechanical packaging required.
b. second source available.
c. logic interface.
d. power supply complexity.
e. ease of maintenance.
f. environment and noise.
g. ease of editing operations.

These"fuzz" factors tend to blur any clear advantage one memory may have over another; namely: price. By price the memories line up as follows (512 7-bit characters):

a.	core memory	$600.
b.	MOS shift register	$250.
c.	mechanical delay line	$175.

In spite of the clear advantage of mechanical delay lines, many manufacturers use core memories, and more and more are using MOS shift registers. These prices are based on a single memory physically packaged with each display. When we consider centralized store refresh memories, things turn out a little differently. Here, we assume that all memory is physically packaged in a central unit for all displays serviced. Certain per unit economic advantages accrue to large aggregates of memory which is another way of saying things are generally cheaper by the dozen.

A complete listing by cost on a per CRT display (512 7-bit characters) basis is now:

a. Individual core memory-------$600
b. Individual MOS shift register---------------------$250
c. Individual mechanical delay line-------------------$175
d. Core memory (16 displays)----$150
e. Disk or drum memory (16 displays) ---------------$100

The important additional fuzz factor here is that these costs are based on full use of the memory. Requirements for fewer displays blur any clear advantage with overhead costs.

Back now to the poor S. B. who can't wait to buy his displays. From his point of view, there are systems fuzz factors that further blur any memory cost advantage. Examples are: Computer interface signals and message structure, or display character size and shape. In spite of the fuzz factors, two things seem clear. First, other things being equal, displays using centralized store refresh memories should now be noticeably less expensive. Second, at the rate which MOS shift register memories are coming down in price, it won't be long before the situation is reversed, i.e. individual refresh memories will be less expensive than centralized memories. How long can you wait?

ENGINEERING DESIGN OF UNISCOPE 100 TERMINAL

K. R. Fechter, Director, Technology and Manufacturing, Communications and Terminals Div., Univac
T. M. Soller, Director, Systems and Software, Communications and Terminals Div., Univac
M. L. Granberg, Director, New Product Development, Communications and Terminals Div., Univac

ABSTRACT

The rapidly increasing need for a low cost alphanumeric terminal in quantity permits several avenues of approach which will result in significantly lower cost terminal systems. Discussed are the requirements of the communication system for various applications and system elements which affect total operating costs, including cost of telephone lines. How the terminal accommodates the design criteria of data presentation, data entry/editing, communications capabilities, physical characteristics, and terminal price, along with hardware techniques permitting a low cost family of terminals to be produced, is also discussed.

INTRODUCTION

A trend is clearly evident that the user of the data processing systems of the future will be interfaced with the system through a multitude of low cost terminals and devices which are closely matched to human impedances. A number of human impedance matching devices are available, but in the foreseeable future the most commonly used for input will be a keyboard of various configurations. The most common output will be printed alphanumeric data followed closely by charts and graphs. Hard copy is an optional requirement.

The requirement for trained operators will diminish, but the ability of the terminal to change its human imedpance characteristic will become increasingly important.

A veritable explosion is taking place in the announcement of new terminal devices for use in data processing systems. The UNISCOPE ® 100 is one of these products and is aimed at a broad class of applications which are primarily communications oriented.

SYSTEMS DESIGN CONSIDERATIONS

The UNISCOPE 100 family of alphanumeric cathode ray tube (CRT) display terminals is intended to satisfy a wide spectrum of requirements for interactive communications-oriented terminals. The following were the five major areas of design consideration from a systems viewpoint:

- Data presentation
- Data entry/editing
- Communications capabilities
- Physical characteristics
- Terminal price.

Data Presentation

The principal elements of data presentation resolve themselves into three factors: presentation quality, format of data presented and quantity of data presented.

In earlier generations of CRT displays, the user was willing to tolerate such negative attributes as flicker, low brightness presentation and ill-designed characters in order to realize the performance advantages of a display terminal. However, the market has matured to the point that the user expects the display to be of high visual quality, with ample brightness and contrast, in a well-lighted room. Careful study of the situation convinced Univac that it was economically feasible to provide a high quality, low cost display. Thus, it was required that the UNISCOPE exhibit no visually discernible flicker or jitter, that the presentation be easily readable in normal office-environment lighting, that there be no confusion among characters within the full 96-character ASCII set, that any intensity variation within characters be minor and that data presented on the screen be readable from at least 7 feet.

From the viewpoint of individual applications, format requirements tend to relate directly to the input or output media employed in the application prior to use of the display. For example, if punched cards were used previously, an 80-character line display is required. If teletype was previously used, a 72-character line is normally required. On the other hand, some applications tend to lend themselves to tabular listings of short lines of data. In such cases, it is preferred to have a large number of short (40-character) lines. Because of this wide variation in application requirements, it was determined that the UNISCOPE 100 must provide a number of format options.

Once a line format is selected, the display capacity is, of course, a direct function of the number of lines. It appears that for many commercial applications, a total capacity in the range of 500 characters is adequate and that the vast majority can be satisfied by a 1000-character capacity, provided the display is designed to facilitate rolling of data. The latter requirement arises from the wide use of displays to access large files of data. Since it is not economically feasible to provide a display of adequate size to view a large file directly, an alternate approach is to use the display as a "window" on the data file and, by software, roll the file past the window. However, because of the potentially large number of displays in a system, it is absolutely necessary that the hardware be designed to minimize the burden imposed on the software by the roll function. This is accomplished in the UNISCOPE 100 family by means of a sophisticated algorithum which permits the deletion of any full line of data and repacking of the format by means of a single computer-generated function.

Data Entry and Editing

From the user viewpoint, it is vital that an interactive terminal have not only the capability of natural entry of data, but also of correction and modification of data. Thus, the terminal must provide to the user a significant editing capability. Two approaches will provide this required capability: (1) sophisticated editing logic can be provided in the terminal and (2) computer software can be used to implement the required capability. Again, the potentially large number of terminals within a given system dictated that, in the UNISCOPE 100, editing logic for the most commonly used functions be included in the terminal in order to unburden software. Editing functions in the UNISCOPE 100 include character, line and display erase plus the ability to insert or delete a character within any line or within the entire presentation format.

User data entry into the UNISCOPE is accomplished by means of a standard typewriter keyboard and/or by means of a 10-key pad. None of the data entered by the user is sent to the computer until the complete entry is displayed and verified by the user and the transmit key is pressed. This design approach not only minimizes the entry of erroneous data, but also permits efficient coupling between the capacity of high-speed communications networks and keyboard entry speed.

Because most manual entries involve a relatively small number of characters, it would be extremely wasteful of communication network and processing capacity if it were always necessary to transmit an entire format of information with every entry. Therefore, the UNISCOPE provides the user with the facility to bracket his entry with special characters, and, thus, only the actual entry will be transmitted to the computer.

Communications Capabilities

Perhaps the single most significant aspect of the UNISCOPE 100 from a systems viewpoint is its communications capability. Many of the available alphanumeric display terminals are displays to which communications capability has been "added." This approach leaves much to be desired in addressing the system problems related to reliable operation of the terminal in a large-scale communications network. From the beginning of its design, the UNISCOPE was considered to be a communications terminal first and a display second. This fact exerted a major influence on many of the detailed design decisions on the UNISCOPE.

The first step in design of a communications-oriented terminal is to define the communications environment in which it will operate. There is a very large number of variables related to the communications environment, and it was determined that the UNISCOPE 100 must be capable of covering a large percentage of them if the desired broad market base were to be established. This is accomplished by means of a number of interface options. However, under all options, the same basic communications discipline is used.

A terminal like the UNISCOPE may be used in either a switched (dial-up) network or in a private line multidrop network. A typical switched network usage is in the time-sharing industry. The advantage of this type of network, from the terminal viewpoint, is that the telephone switching network resolves contention situations. That is, only one terminal can be connected to a given system channel at a given point in time. Other contenders for the channel are queued by the busy signal until the channel is freed. In applications such as reservation systems, where each terminal is on-line for a high percentage of the time, economics usually dictates use of a multidrop private line. This type of operation is much more demanding of the communications discipline than the switched network case because it is necessary to provide techniques for contention resolution and addressing in the multidrop environment. The UNISCOPE 100 communications discipline permits operation with both switched and multidrop private line networks.

The UNISCOPE 100 terminals may connect directly to the communications network or interface through a multiplexer. The use of the multiplexer provides three significant system advantages where multiple terminals are required in a single location. First, it reduces the cost of the communications network by eliminating modems. Secondly, in a switched network, it permits more efficient utilization of the communications system in that an individual display would normally require only a small percentage of the available baud rate on a high-speed line, whereas a number of displays multiplexed to a single line provide a better match to available capacity. The third factor relates to the fact that the UNISCOPE 100 communications discipline is based upon polling; that is, data transfer from a terminal to the computer occurs only when the terminal has data to transmit and the computer polls or requests data from the terminal. There are two types of data polls used with the UNISCOPE 100: specific and general. Use of the specific data poll requires the computer to individually access each terminal or terminal component. The general data poll permits the processor to elicit a response from any terminal or terminal component which has an input waiting and whose address is included within the generalized portion of the poll address. The addressing structure used in the UNISCOPE 100 is a three-tiered structure in which, effectively, the first tier pertains to the multiplexer, the second to the display terminal and the third to display terminal components. If a multiplexer is addressed specifically and the rest of the address is generalized, the multiplexer will permit the highest priority (defined by hard wiring within the multiplexer) display terminal with a pending input to respond; if no terminal connected to the multiplexer has a pending input, a single no-traffic response is returned. The systems advantage of the multiplexer in conjunction with the general poll is the facility it provides for limiting the number of poll-points and thus polling overhead in the system.

Throughout the previous discussion, the UNISCOPE 100 has consciously been described as a display terminal. It was recognized early in the design of the UNISCOPE that there are many devices which, when used in conjunction with a display, naturally complement and extend the capability and applications potential of the display itself. These devices include printers, magnetic tape cassettes, and punched card readers. It was also obvious that the terminal provided sophisticated communications electronics and a small high-speed memory which might be useful to a number of types of equipment requiring low-cost remote computer interface if the unit were designed to facilitate access to these elements; thus, an auxiliary (eight-bit, parallel, half-duplex) interface option is provided in the UNISCOPE 100. The auxiliary interface, which permits connection of 12 terminal components, indeed permits the UNISCOPE 100 to function as the center of a remote terminal complex.

Figure 1 reflects the relationship of the various components of the UNISCOPE 100 in a small-scale private line application. The communications line discipline permits connection of approximately 1500 terminal components or devices to a single private line. Furthermore, the line discipline has been carefully designed to permit intermixing of compatible batch terminals on the same line as the UNISCOPE.

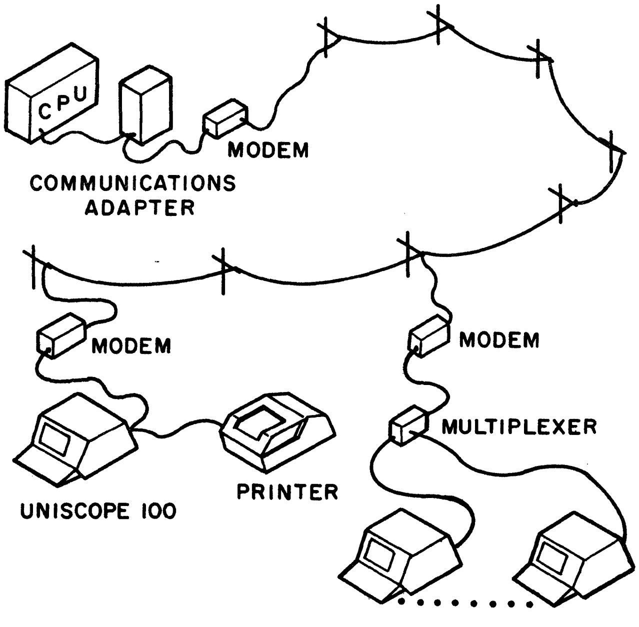


Figure 1. Small-Scale Private Line Application

Physical Characteristics

Careful study of user requirements convinced Univac that for market acceptance of the UNISCOPE, it must be physically small. The general guideline established was that the device be in the general size range of an electric typewriter. Although this requirement imposed severe design constraints, the goal has been fully met (figure 2).

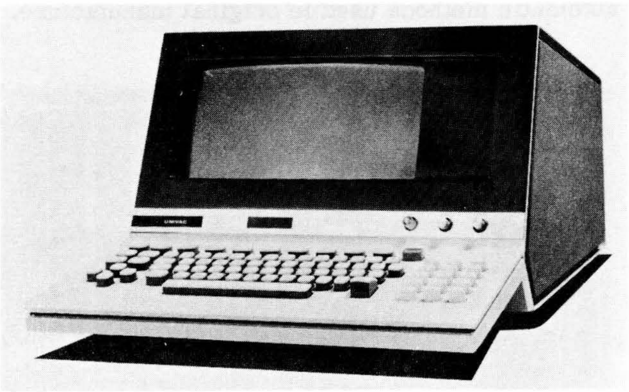


Figure 2. UNISCOPE 100

Terminal Price

Of course, for a display terminal to receive wide market acceptance, it is necessary that it be fully competitive on a direct comparison basis with other forms of remote computer terminals in addition to being priced competitively with other displays. Studies of the marketplace indicated that a monthly rental base of $75 per month including maintenance has been effectively established by the teletype terminal. It should be noted that this goal had to be met without use of such pricing artifices as offering a nonviable configuration at this rental figure. The UNISCOPE is indeed offered in a fully viable configuration at $75 per month. Furthermore, the unit's communications design permits the unit to be used in a straightforward fashion without requiring expensive "black boxes" to permit its use in a system.

HARDWARE DESIGN

The hardware was designed to incorporate what is generally considered to be high performance specifications. These can be summarized as follows:

1. Brightness: a high efficiency phosphor (P-31)

2. Flicker: the display is refreshed at 60 Hz (50 Hz where prime power is 50 Hz)

3. Characters: the stroke technique used is capable of resolving a full ASCII set on a unique basis

4. Memory: a temperature compensated coincident core memory

5. Maintainability: easy to change pluggable modules

6. Solid state: the only vacuum tube is the cathode ray tube

7. Magnetic deflection: no high voltage coupling to electronics required.

Power Considerations

The UNISCOPE 100 was designed using low power techniques. Power consumption is reduced from previous methods by reducing the voltages required and by consuming power only when necessary. The unit uses approximately 100 watts total.

Memory

To conserve power in the core memory, approximately 4 volts is used in the metering voltage for the drive current as contrasted with a typical 15 to 30 volts used in more conventional systems. To achieve equivalent metering accuracy so that "worst case" design criteria can be met, a high performance regulator is used in each of the current metering circuits.

The complete memory (figure 3) is contained on one 9- by 12-inch printed circuit card. The memory contains 8000 bits organized in 1000 eight-bit words and uses a maximum of 10 watts.

Deflection

The UNISCOPE 100 utilizes a single-ended flybackless deflection system. Power consumed is proportional to the distance of the beam from the center of the screen. This system is contrasted from a push-pull deflection system which would use approximately twice the power. Additionally, the voltage needed to drive the yoke is proportional to the required deflection rate. By eliminating the need for flyback, the deflection velocity is considerably minimized. Figure 4 illustrates how the programming of the deflection system eliminates the flyback requirement.

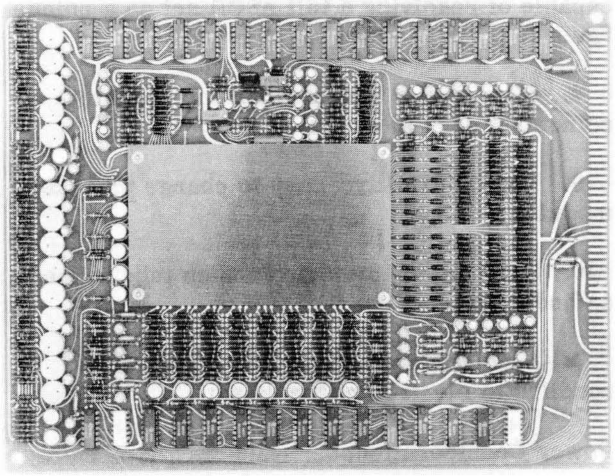


Figure 3. Memory Card

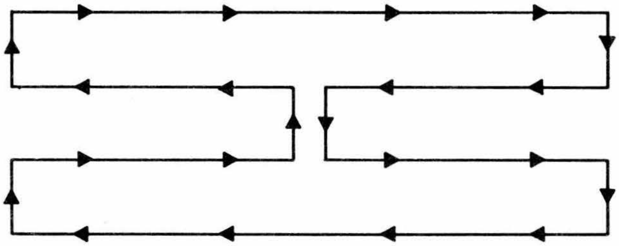


Figure 4. Scan Method

An operator typing information from the keyboard would not detect that lines are refreshed from both directions in that the data appear on the screen in normal typewritten fashion.

Focus

To assure proper ratio of line width to character size, a beam width of 10 to 15 mills is required. A low voltage electrostatic focusing of the beam was developed for several reasons. First, magnetic focus requires heavy coils which, secondly, require extensive attention to mechanical mounting. This total assembly is costly although certainly adequate from a performance point of view.

MAINTENANCE CONSIDERATIONS

A functional module design approach was used to minimize repair time and cost in the field. Each functional element, i.e., memory input/output, deflection, keyboard, etc., is a replaceable pluggable module. Identifying the suspected failed module is reduced to a few major elements and requires relatively little intimate knowledge of the total machine.

This dictates the need for some large functional elements, as shown in figure 5. Suspected failed modules will be repaired at depot level by the automatic methods used in original manufacture.

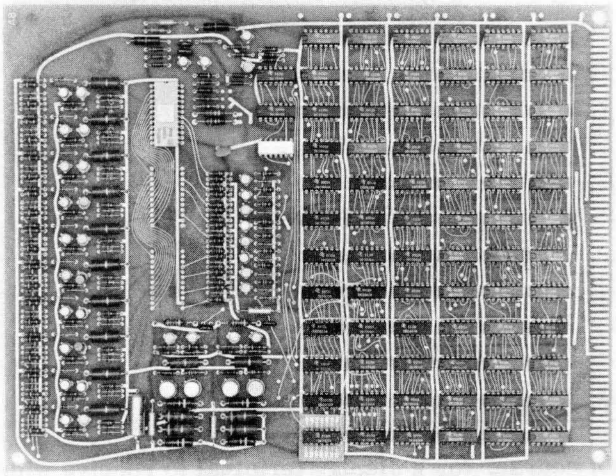


Figure 5. Logic Card

MANUFACTURING CONSIDERATIONS

The difference between success and failure of this product, irrespective of performance, will be price. To achieve low price, maximum use of automation, standardization, and large production runs are basic.

The use of large function cards, which makes isolation of a trouble in the terminal to the specific module an easy task, requires very sophisticated test equipment in order to isolate the trouble to the failed component in the module or logic cards. Manual test equipment would be too time consuming; the logic cards contain as many as 120 integrated circuits. Most of the circuits now used are in the DTL 930 integrated circuit family. These cards are tested by using the brute force approach of checking the output of every logic node with permutated inputs. The test is completely under computer control. The basic program is sufficiently flexible so that if a new failure mode in the card were discovered which the present program could not isolate, the program could be modified so as to readily isolate the problem. The testing time on each card is less than 1 minute, and on the average, each card requires approximately 1000 tests. If there are faulty components on the card, a printed copy stating which component should be replaced is attached to the card at the end of the test. The card is not tested from the connector because all of the logic nodes are not available. The card is probed on the surface with 500 to 600 electrical contacts.

Low cost 930 series integrated circuitry is used for most of the logic application. Because of the functional partitioning of the logic, it would be easy to use larger scale integrated circuitry in this area if the 930 series circuitry were to become uncompetitive.

FLEXIBILITY IN LOW-COST INTERACTIVE COMPUTER TERMINALS

Donald R. Haring, Computek, Inc.

ABSTRACT

There is an ever-increasing use of interactive computer terminals in all phases of society having computer applications. The requirements of these various applications are as varied as the applications themselves. Even in a given user community one finds great diversity in the requirements. Furthermore, a given user's requirements do not remain static.

We conclude that the ideal terminal design must allow the terminal properties to be so configured as to closely match the user's requirements, even when the requirements change. However, the desired flexibility must be provided at no appreciable additional cost. This paper describes an approach to the design of an economical, flexible, interactive computer terminal. Although we shall concentrate on the design of cathode-ray tube (CRT) graphic terminals, this approach is also applicable to a much wider spectrum of applications.

INTRODUCTION

A computer terminal is a computer with special input-output devices to interface man and machine. The usual approach taken in designing a computer terminal is to (1) define the requirements as well as they are understood at design time, and (2) design a special piece of hardware; i.e., a special-purpose computer, to satisfy the requirements. The special-purpose computer approach is used in an attempt to produce an economical terminal. At one time, this same approach and reasoning were used extensively in the design of mainframe computers destined for specific applications. However, such is no longer the case. Most computer systems designers have now turned to the programmed general-purpose computer to implement the majority of their systems requirements. It was found that special-purpose computers were not as economical as programmed general purpose-computers because each new set of specifications required a completely new design, and because the special-purpose designs did not lend themsleves to changes created by the differences between the supposed requirements at design time and the actual requirements at delivery time.

DISCUSSION

Since the computer-terminal designer is faced with many of the same problems that face the computer designer, it would seem reasonable that the arguments for the use of programmable general-purpose computers in terminals, as opposed to fixed special-purpose computers, follow the lines of the computer designer. However, until recently the hardware cost of such a programmable computer in the terminal was prohibitive, because of a certain amount of hardware overhead. With the cost of integrated-circuit logic now dropping so dramatically, it is now practical, with proper design, to consider a terminal with a programmable computer. This design must take into consideration the environment within which the terminal computer must operate.

The major task of an interactive computer terminal is the transfer of information among the several subsystems of the terminal. Only the most trivial modifications to the information are required. Thus, the architecture of the terminal computer should be geared to handling information transfer. The data-bus system, widely used in the input/output system of many computers, is particularly efficient, in terms of hardware and interconnections, at this task. In this system, all devices are attached to a common data bus and are identified by unique addresses, which are also sent over the data bus or a second control bus.

Let us now consider the CRT graphic terminal. The major subsystems of the usual graphic terminal are: (1) the display module, (2) the manual input devices, (3) the character generator, (4) the graphics module, (5) the computer interface, (6) the local memory, and (7) the control element. Except for the first two subsystems, a programmed general-purpose computer can function as the major component of the terminal subsystems. (As a matter of fact, the first computer CRT displays operated in precisely this way. However, the demands of the graphic operations make the total computer system grossly inefficient. Most of the graphic operations are simple but repetitious, and thus make severe demands on computer time. This fact led to the design of special computer CRT terminals based on a special-purpose computer).

Assuming that a general-purpose computer is a component of the terminal, the designer's problem is to make a trade-off between programming all functions of subsystems 3 - 7, and supplementing the computer with special hardware that will function as some of these subsystems. The trade-off is normally reduced to a consideration of two related questions: (1) the amount of graphic information that must be displayed within a given time segment, and (2) the cost effectiveness of read-write memory versus read-only memory in a given subsystem.

With these thoughts in mind, let us describe the design of a low-cost, table-top CRT terminal, to indicate how the final configuration was selected. The basic specifications of the terminal are as follows:

(1) Refreshed CRT-1000 character capacity
(2) Keyboard input
(3) 64 ASCII characters
(4) Alphanumeric only
(5) Serial or parallel computer interface
(6) At least 1000 character local storage
(7) Local editing required

The core of the terminal design is a self-contained, 10-bit byte oriented, microprogrammed general-purpose computer whose architecture is centered around data and control buses. The computer has only three classes of instructions: (1) transfer data from element i to element j, (2) test if bit k of the data bus is 0 or 1, and (3) send a specified 10-bit binary number to element i. The first class of instruction allows the data to be modified in very simple ways (e.g., the data can be incremented, complemented, masked, etc.) during the transfer process. Instruction time is 1.5 microseconds. Programs to run the computer are normally contained in low-cost read-only memory. To reconfigure the terminal functions, the contents of the read-only memory are changed.

Any operation in the terminal is done by a microprogram. For instance, an edit routine to delete a character on the screen is a microprogram which operates roughly as follows:

(1) Find the character that is identified as the one to be deleted, and remove that character code from memory.

(2) Move the next character into the space just made by deleting the character.

(3) Repeat step (2) until the end of the line is reached.

Clearly, the delete-character microprogram is used as a subprogram for the delete-line microprogram. Not only are programs shared (thus read-only memory), but some registers and other components in the terminal are shared for several tasks.

With the addition of the proper read-write memory, this computer can do all the tasks for subsystems 3 - 7. A serial read-write memory is used because the basic refresh required allows this low-cost memory to provide the desired performance. To simplify the microprograms, the serial memory is operated in an associative mode.

To ensure a fast reaction time to user edit requests and high-speed computer input-output, some cost-effective supplemental logic has been added to the terminal. First, a low-cost dot-matrix character generator with its own read-only memory was added becuase the high-speed repetitious requests would either degrade response time or demand a higher speed read-only memory for the microprograms. Second, when serial connection to the central computer is used, a special serial input-output subsystem is used, for much the same reasons as for the use of the special character generator.

The self-contained computer provides an "intelligent" terminal, thereby reducing its time and memory demands on any central computer to which it is attached. The self-contained computer also allows the terminal to act as a standalone data-entry station. Such a terminal can readily service peripherals connected to it, such as hard-copy generators, badge readers and tape cassettes. Finally, it is much simpler to create off-line test and diagnostic equipment, thus reducing the cost (in both time and money) of maintaining such terminals.

CONCLUSION

In closing, let me add that the majority of users do not care whether the terminal contains a computer or not. They are not interested in elegant hardware and software, but in results. Therefore, the computer aspects of the terminal should normally be suppressed. The user should have to operate only those controls necessary to run the terminal, and not a computer panel with a bank of lights and switches which serve only to confuse him.

DATA BASE INPUT AND TEXT HANDLING IN AN OCR SYSTEM

Gerhard Reitz, MACRODATA Company, Chatsworth, California *

ABSTRACT

This report deals with the input to a total text processing system. The input, read by an Optical Character Reader (OCR), is typed on two different legal-size forms of a variable number of pages.

Since production-oriented OCR systems are new, special emphasis is given to experiences in forms design, general considerations of multiple font optical character readers, and, especially, the software aspect of processing raw text, including the determination of documents, fields, the extent of sentences, etc. All programming (except input) is done in SNOBOL4. The OCR is of Philco-Ford design.

INTRODUCTION

The desirability and potential of using automatic data processing techniques and equipment for the processing and analysis of large volumes of textual material has long been recognized.

Numerous experiments have demonstrated the validity of various automatic text manipulation techniques. These were usually of a research nature and disregarded any input problems. Economic feasibility (and operational constraints) however, require practical input devices which automatically convert text into machine-readable form. This requires the integration of research techniques, hardware, software, and user needs into a viable system.

The overall AUTOEX project[1], of which this work is a part, has as its primary objective the design and implementation of a text processing system which:

(1) employs an optical character reader to convert typewritten, formatted and unformatted text into computer-readable magnetic tape;

(2) integrates state-of-the-art computational and linguistic techniques into efficient content analysis modules and supporting programs;

(3) is implemented in a high-level programming language (SNOBOL4);

(4) runs on third generation hardware processors (GE635, IBM360, UNIVAC1108, etc.).

The AUTOEX system has been implemented and has demonstrated technical feasibility.* Tests for determining its operational potential are now in progress.

Addressing ourselves here to the problem of data base input, we might ask if Optical Character Readers have finally delivered us from the clutches of keypunching. We can answer this with an unqualified yes: faster typists can now feed an expensive piece of hardware, but added software is needed at the front end.

This report will go into the details of unfamiliar procedures and will gloss over some of the idiosyncrasies of the particular implementation, while giving special emphasis to the text processing aspects. (Only few papers have covered this in detail[2-6]).

The system is designed to process variable 8-1/2 x 14" page reports, dealing with various political and military areas of interest. They are typed in different locations on typewriters of one font group. The reports, read by a Philco Ford Optical Recognition System[7], are written out on magnetic tape and fed into a GE 635 or IBM/360, Model 40 or above, where they are processed, indexed, and put into a format acceptable by a minicard or FMA storage and retrieval system. All the software, except for input and conversion routines are written in SNOBOL4, version 2[8].

Four phases will be discussed:

- o Document design and preparation
- o OCR considerations
- o Raw text processing
- o Results

---

*The work reported here was done while the author was at the Bunker-Ramo Corporation, Canoga Park, Calif.

* This work was supported in part by the United States Air Force, Rome Air Development Center, Griffiss Air Force Base, New York (Contract AF30(602)-69-C-0197).

A. DOCUMENT DESIGN AND PREPARATION

Document design requires detailed knowledge of the OCR programming features, as well as the goals of subsequent text processing, an excellent printing firm, and meticulous care. Even then, only our third redesign (and printing) was correct.

Where necessary, the form can be broken into arbitrary fields with or without the use of printed (or typed) control symbols. In any case, various instructions have to be given via the stored OCR program, to position the scanner correctly. Even though we are concerned here with a multifont (but fixed spacing) reader, the exact group of fonts has to be communicated through a program instruction. In addition, either a normalization constant has to be included, or else an initial capital letter is needed to let the scanner perform this function.

Where printed verticals, boxes, or text may interfere with typing (such as for indications of the right margin), violet non-OCR ink should be considered, since black lines may have special functions. In this respect, uniform typewriter tab settings for all pages should be kept in mind. Mark-sense boxes can also be designed, where the X's will be read as bits and packed four to a byte. A read command with several parameters permits checking each field for a specific number of characters and for other special conditions (numeric, alphabetic, special characters). A disallowed character will be read as a "confusion" character, or may be programmed to reject a page. (A special switch controls the automatic rejection of a page after five consecutive "confusion" characters. In a multi-page document, this will of course wreak havoc with the page coordination!)

Pages with a read area larger than 7 1/2 x 7 1/2" can be advanced by programming, and can undergo a second scan, thus enlarging the area to 7 1/2 x 13". This will necessitate printing a special mark, in addition to the benchmarks at the top of each page, which enables the scanner to detect any skew.

In our case, an additional switch is available, which enables the machine to recognize and output each space, once the first non-space character is read. In the off-position, a string of non-initial spaces is collapsed into one space. As discussed in the section on text processing, this feature is quite important for the characterization of the type of sentence. Even so, any indentations for paragraph indications are lost, a regrettable sign of non-cooperation of hardware designers and analyst, especially since this was pointed out very early.

In order to indicate the complexities of OCR programming (over 1K instructions in the 4K OCR memory) and text-processing software, I should mention that the heading page consists of 19 fields from 1 to 27 lines deep and 24 - 78 characters wide, while the (optional) continuation page(s) contain three fields, the largest 63 lines deep and 76 characters wide. On each page, only one field is known to be numeric and one alphabetic; the others are mixed and, with one exception, free form.

No matter, how well designed a document, it is the typist who gives it the last "touch". A standard joke refers to our mint-condition documents, a result of our Typist Guide.

This guide starts out by stressing the importance of vertical and horizontal alignment of the document through typing a capital letter into two boxes, provided in the top right and left hand margin. From this moment, the paper should not be removed from the typewriter until completely finished using only the approved line-spacing. For the multi-page reports, with which we are concerned, the page number field is filled out in the following manner: 1 of 1 2 3 4 5. At the end of the report, the typist just brushes over the not applicable numbers (with SNOPAKE, Liquid Paper, etc.). Other corrections also are best taken care of in this manner, or by means of overstriking them with KO-REC-TYPE, since even good erasures change the reflectivity of the paper and smudge easily, conditions to which the OCR is very sensitive.

Of course, corrections, or composition of letters by overstriking or underlining (ø ≠ ! $̸ NØ strike -øu̸t̸$) result in the output of a "confusion" character. Crowding or squeezing letter is also dangerous. If underscoring is absolutely necessary, it can be performed with a special non-OCR readable ball point pen. If at all possible, a new field should be started with an initial capital letter (e.g. on a continuation page).

Superscripts and subscripts, such as $Al(OH)_2^+$ are not acceptable, unless the OCR is specially designed to read such small type.*

Again, for programming reasons, dates should be handled in a uniform manner (4/14/1969 or 4 April 69, etc.). It should be kept in mind that hyphens at the end of a line will be removed by programming.

It is by now obvious, that the typewriter must be kept in excellent mechanical condition. The slugs must be cleaned often, the fabric ribbon should not be frayed (one-time carbons are best); pressure should be decreased on new ones, and increased with use. Character misalignment, sticking keys, broken slugs, must be corrected immediately. For this reason, it might be well to type the typewriter serial number in a special field, so that immediate identification can be made, if consistent errors turn up.

Since these pages are fed into the OCR at a rapid rate, the top edge is especially sensitive to creases, and the document may not be stapled or folded. It also must be kept clean and smudge-free.

---

*Footnotes such as this are difficult to handle by programming and should be avoided by incorporating them right in the text.

B. OCR CONSIDERATIONS

Among the less obvious of the 38 commands available to program the OCR, one can be used to step an internal counter for counting pages in multipage reports, etc. An accumulator allows BCD addition and subtraction to cross-check or verify totals and can be output separately to typewriter or magnetic tape. Four toggle switches can be sensed by the program to effect special branching. Output can occur to typewriter or magnetic tape (6-bit IBM tape format, 200 or 556 cpi, with lateral and longitudinal parity).

The following special code assignments are important for their repercussions in further text processing:

(1) A semicolon is always coded as a colon (a not very happy solution to a hardware problem).

(2) When, in a given font, a lower case el and a one are identical, this character is considered a one in a numeric or mixed field, and an el in an alphabetic field. If the characters are unique, they are assigned the proper code.

(3) When, in a given font, an upper case Oh and a zero are identical, this character is considered a zero in a numeric or mixed field, and an Oh otherwise. If the characters are unique, they are assigned the proper code.

(4) For upper case alphabetic letters, a two-character sequence will be written, the first being a non-latching shift character. All other characters will be assigned just one code. (The prototype hardware, recognizing many more characters as well as underscores, used four escape codes, one for each character, thus filling up memory much faster, and requiring more preprocessing).

C. RAW TEXT PROCESSING

The raw text, output by the OCR on magnetic tape, is now read by the general purpose computer. The following five objectives have to be met to allow this raw text to be processed by the AUTOEX indexing system.

- o Determination of each field
- o Determination of the extent of a document
- o Processing a line of text
- o Determination of the extent of a sentence, and
- o Preparing the sentence for further processing.

(1) Determination of Each Field

The maximum extent of a field is determined by its beginning and ending columns, the number of lines, or its x-y coordinates; however, the end of a field can also be forced by typing a + as the final character. In our case, a field count is necessary to determine the type of page, and thus the beginning of a document.

(2) Determination of the Extent of a Document

For regular-sized, one-page documents, this condition can be satisfied by reading in a record. For legal-size, variable page documents, this may not be a trivial process. (It is important to recognize that an end-of-page mark follows the last character, when the memory buffer is output; this variable record can be read under GECOS or OS in an undefined format; this unfortunately does not work correctly with the Fortran I/O routines that SNOBOL uses).

Usually several conditions need to be recognized in order to determine the type of page and, therefore, the extent of a document; they are characterized by the buffer overflowing in specific fields, causing automatic output.

(3) Processing a Line of Text

A line of text ends with one of the following characters: end-of-field, end-of-page, or carriage return. Processing essentially consists of analyzing the characters in each word, and converting any shift code(s) into one of the five capitalization codes, which are appended to each word: (a) all lower case; (b) all upper case; (c) initial upper case; (d) all numerics; and (e) mixed cases. For the determination of all capitalization codes except (d), numerics were disregarded, such that "DC-3" would have an all upper case code appended.

This classification of words incorporates all the necessary information for the subsequent determination of sentences, and also makes a dictionary lookup possible which could differentiate, if necessary, between GaS and gas, CAT (Clear Air Turbulence) and cat, CARE and care, etc.

At this point, the zero and el disambiguation takes place. The processing of these characters, which may stand interchangeably for zero/Oh or one/el, went through many changes, the last of which finally turned out to be also the shortest. Even though some counter examples can be made (where aren't they possible in language data?), short of listing the exceptions, the following ordered decision table was almost perfect:
(See Figure 1)

(4) Determination of the Extent of Sentences

By a sentence, we really mean a unit of information, such as heading, titles, footnotes, legends, and ordinary sentences. In a columnar (tabular) arrangement, such as a price list, each line is arbitrarily defined to be a sentence. For purposes of further processing, six sentence classes are established (see next page).

FIGURE 1
DISAMBIGUATION TABLE

Input	Vowel in Word	First Condition	Second Condition	Decision	Example
0	No	word length =1	-	Capital Oh	0
1				One	1
0	No	word length ≠1	At beg. of word	Capital Oh	Owl
			Not at beginning	zero	80,000
1	No	≠1	-	One*	F111; 171
0	Yes	a $ preceding	-	zero	US$80
1	Yes	a $ preceding	-	One	US$18,111
any	Yes	any-./ in word	-	Let this character break word in two. Go to next rule.	BOS/360 F-111
0	-	word otherwise all alphabetic	-	Capital Oh	DOD
1	-		-	el	also
0	-	word otherwise all numeric	-	zero	700
1				One	1231
0 1	-	word mixed	-	(special cases to be tested for)	

* May give the wrong decision for some abbreviations.

A critical point in this sentence recognition concerns the arrangement of the text, which should be presented in its original form, especially as regards to underlines, single and double quotes, and the presence of spaces (even if they are not counted exactly because of hardware restrictions concerning different font sizes). The problem is identical to the one of sentence recognition in machine translation, where we had accumulated extensive experience. It can be stated succinctly but trivially:

Periods or spaces may end words or sentences (the role of colons and semicolons is not considered here).

In the following 12 rules are enumerated. They are incorporated in the SENTOR (Sentence Recognizer) Program in SNOBOL4; for the greatest programming efficiency, they are, however, intertwined with the paragraphing rules, the dehyphenation rules, and the rules governing the conjoining of pages.

(4a) General Rules

Rule 1: In a single-spaced document, an end of sentence is forced before a double spaced line.

Rule 2: An end of sentence is forced before each new field and before the end of the document.

(4b) General rules for periods

Rule 3: A period, not followed by a space, cannot end a sentence. e.g., 6.67/7.62 meters A.S.I.S.
Rule 4: A period, followed by two spaces and an (initial or all) capitalized word, ends a sentence. e.g., ... map is 1: 800,000. Following this...
... by April 14, 1969. Some taxpayers...
Rule 5: Any period inside brackets cannot end a a sentence.
e.g., ... (Vol. XXXII. No. 8), by ...
... (Ref. 1, pp. 65). ...

This empirical rule was never violated in the documents processed.

(4c) Rules for enumerations and abbreviations

We consider an enumeration itself an inert sentence introducer and drop it from the beginning of the sentence, so that the parser program later does not have to concern itself with forms like (4a), 4A., 4.a., IVa, (A.4.), C11., etc.

Rule 6: It is paraphrased here, short of flowcharting it: Strip off any parentheses and/or ending period; if no period is found inside the word, see if it is a Roman or arabic number, or a single letter, or an allowable combination of these, otherwise let the leftmost period break the word and check each part for the above again.

Rule 7: The following unambiguous abbreviations are followed by a number, and therefore cannot end a sentence:
Ref. Vol. Nr. No. p. pp. Jan. Feb. ...

Rule 8: The following abbreviations denote titles, and therefore cannot end a sentence.
Prof. Dr. Eng. TSgt. ...

Rule 9: If an ambiguous abbreviation (i.e. an abbreviation which may or may not end a sentence) is followed by at most one space and a comma, colon, semicolon, or quote, or if it is followed by one space and a lower case word (not a number), the period is not an end of sentence.
E.g., Ibid.; ...175 h.p. at 300 rpm. This ...
... page 206 ff.: ... published by A.N.A.S. and ...
... Jefferson Place, N. W., Washington, D. C.

Rule 10: Periods used in the following context (personal names) do not end a sentence: Abbreviated or spelled out titles or the word 'by', optionally followed by a parenthesized expression of up to three initial capitalized words, followed by either an initial capitalized word (or letter with period), or two capitalized letters with periods, or one capitalized letter with period and an initial capitalized word, or one initial capitalized word and a capitalized letter with period, followed by an initial capitalized word or an all capitalized word, optionally followed by an all capitalized word or a hyphen and an initial capitalized word or a hyphen and an all capitalized word.

Obviously, a graphic representation is not only much better, but also more accurate; thus, { } denote alternatives (logical OR), + denotes concatenation with one space, Null denotes an empty string, and quotes surround literals. An initial capitalized word is indicated by Word, an all-capitalized one by WORD, and a single capitalized letter by L. An improved, but untested version includes over 65 combinations!

E.g., Mr. Jacques H. BOITREAUD
Colonel D.I. Munoz-Delgado
Major General (Generalmajor) Kurt SCHLAUBERGER
... by D. R. Auld. ...

All the sentences identified with the above rules receive the classification S (normal sentences), and are passed on to the indexing and the parsing programs.

(4d) Rules for sentences without periods

Rule 11: Determining headings and titles, requires finding, on a new line, an arbitrary number of spaces, followed by an initial or all capitalized words (they may be preceded by a number), followed by a space, a colon, a quote, a hyphen, or a space and hyphen, and then followed by spaces to the end of the line.
E.g., REFERENCES
Comments:
Table of Contents -

These sentences are classified by a T(for Table of Contents), an R (for Reference(s)), or an H (for Heading, such as the above Comments). Those with a T or an R are concatenated with all the text to the end of the document, and passed on to the indexing, but not the parsing program, since this was empirically found to work correctly.

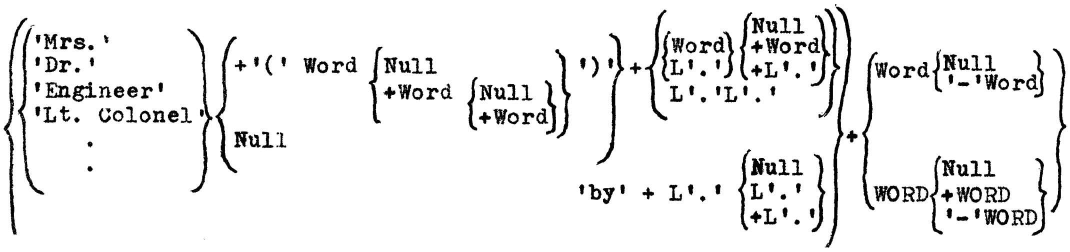
Rule 12: This is concerned with columnar (tabular) data, where every line is examined for one or several strings of spaces (each more than two spaces long), repeating at about the same location in each line, with the exception of a continuation of a column on the next line. Each such line is considered a "sentence", classified as type C (columnar), and passed on to the indexing program, again bypassing the parser.

(5) Preparing the sentences for further processing

As explained above, all sentences receive one of six classifications which determine their further processing. In addition, they are typed with a P if they occur at the beginning of a paragraph. This is necessary for the indexing process, where the application range of certain index modifiers is restricted to the same paragraph in which the prime index word appears.

FIGURE 2

IDENTIFICATION OF PERSONAL NAMES



Also, the sentences of each document are counted so that the indexing program can fill up the available array space in an optimal manner. They are output onto disk as strings (no other form is allowed) although we would prefer to process each sentence as a word array.

D. RESULTS

Many timing runs have been made, but due to the varying environment, they are not strictly comparable.

On an IBM 360/65 - 40 ASP configuration with 310,000 bytes available to SNOBOL 4, 8 documents (= 20 pages or 87 sentences) were run in 108 seconds, or 1.24 seconds a sentence. This included input from magnetic tape and one type of output.

On a 360/65 with MVT and 250,000 bytes available to SNOBOL, 10 documents (=27 pages or 351 sentences) required 375 seconds, or 1.07 sec/sentence. Another run with the same configuration, for 77 pages or 619 sentences, lasted 930 seconds, or 1.5 sec/sentence. These last two runs required three types of output and considerably more garbage collections because of the smaller core space available.

All the textual processing gave very satisfactory results.

Improvements for run-time efficiency were easily programmed in SNOBOL4, an ideal language for this kind of sentence processing, as well as the subsequent indexing, word concordances, and document accession lists.

In the light of information on the functioning of internal algorithms, the program was shortened by about 100 statements; the processing of a few special sentences, occurring in many documents was suppressed, as was the output of capitalization codes (except for further processing by the word concordance module); additional I/O modules were added to create variable length records (the length of a sentence). An additional task was taken on by scanning each sentence and replacing the ten most frequently used words by codes.

This FORSENT program, combining the previous SENTOR with the pre-processing FORMATTER programs now take, on the same configuration, .9 seconds/ sentence.

Later on, sample tests were conducted with a compiled version of SNOBOL4 (instead of the interpretive version 2), produced by Prof. R.B.K. Dewar at IIT, Chicago. They showed a speed improvement by a factor of more than ten, as well as a core size reduction by one third. The latter will improve the speed even more because fewer garbage collections will be necessary.

REFERENCES:

(1) "Optical Character Reader (OCR) Applications Study", Final Report, Contract F30602-69-0197 Bunker-Ramo Corporation, April, 1969.

(2) Oomen, U., Automatische Syntaktische Analyse (Mouton & Co., The Hague, 1968).

(3) "On the Recognition of Personal Names in Natural Text", E. B. Altmann, June 29, 1967, IBM Research Paper RC-1871

(4) "An Algorith for the determination of End-of-Sentence in Continuous Text", E. B. Altmann, June 1966, IBM Research Note NC-618.

(5) "A System for Automatic Recognition of Names of Persons in Newspaper Text", C. G. Borkowski, March 3, 1966, IBM Research Paper RC-1563.

(6) "The Application of Print Readers to the needs of Intelligence Agencies - A preliminary survey of present Machine Capabilities", J. W. Schwartz, March 1964, Institute of Defense Analysis, IDA/HQ 64-2553, AD-440 749

(7) Program Instruction Manual for OCR (Optical Character Reader) June, 1968, Philco-Ford Corporation, Willow Grove, Penn.

(8) Griswold, R. E., The Snobol 4 Programming Language (Prentice-Hall, Inc., Englewood Cliffs, NW, N.J., April, 1969).

THE USE OF APL\360 IN MECHANICAL ANALYSIS*

Alan L. Jones, IBM Systems Development Division, Endicott, N. Y.

ABSTRACT

The availability of the *APL*\360 terminal system has greatly affected the manner in which engineers approach problems. The basic character of APL is described with emphasis on those aspects which are helpful to the engineer. Mention is made of the use of APL in the following areas:

- solution of differential equations
- curved elastic bar analysis
- impact printing
- dynamic buckling of thin rods
- contact stresses in layered elastic solids
- computation of hypergeometric functions
- holography
- fast Fourier transform
- numerical quadrature
- interpolation
- statistical programs
- complex arithmetic
- plotting of results

INTRODUCTION

While the computer has had a vast effect on the method with which the engineer may attempt to solve a problem, its main handicap is that it excludes the engineer from the solution loop. Too often, valuable time is lost because the engineer may not gain access to the computer when he needs it; when he does get computer time, he may find that his program has not been "debugged" properly, or he may discover that some of the parameter values in his program must be changed. This time-consuming operation can become frustrating to an engineer. Fortunately, there is an answer to the problems described thus far. The solution is to bring the computer to the engineer and give him a programming language uniquely suited to his needs. Now, the engineer can formulate his problem and dial up the computer from his desk. He writes the program, but he does it right from the typewriter of a terminal. If his program needs changing, he can edit within a statement and soon has the program running. The output is immediately visible, and the engineer may change parameter values as needed. This paper describes the manner in which the Mathematical Sciences Department at the IBM Systems Development Division in Endicott has used time-sharing over the past number of years. In particular, it focuses on the use of the *APL*\360 system.

USE OF QUIKTRAN AND APL

Six years ago, our department solved problems in the tedious manner described in the first part of the Introduction. Then, the department personnel were introduced to TELTRAN, the predecessor to QUIKTRAN. Almost immediately, the method of attack of problems changed. Whereas much time had been spent writing quite complicated FORTRAN programs which could handle every eventuality, programs became very simple yet efficient. For instance, in the early stages of the solution of a problem, no provision would be made for terminating a program. The person at the terminal would merely observe the output and, when he had seen enough, he would interrupt. Also, a lot of time coding and checking was saved since many programs could now be composed right at the console. It tended to make the programs initially less accurate, but it was a very productive lack of accuracy. The hours spent at a desk, checking a program over and over, were replaced by immediate error detection and correction at the terminal system.

Some of the problem areas worked on during the days of QUIKTRAN were:

- flow of blood in a tube
- analysis of the dynamics of a column under axial load
- anisotropic stress analysis
- wave propagation in pneumatic tubes.

Then, in 1967, the department was introduced to the *APL*\360 system which was operating at the IBM Research Center, Yorktown Heights. The system was developed at Yorktown Heights and is modeled after the language described by Dr. K. E. Iverson in his book, A Programming Language.[1] After a short time, the entire department switched from QUIKTRAN to APL. It was immediately evident that the conciseness and power of APL made it an ideal terminal language. Now a single statement could replace many statements in a FORTRAN program. This conciseness is due to the fact that APL is based on the concept of the vector, that is, an array of numbers.

---

* APL\360 is available from IBM as a program product under a modified license agreement.

If one wishes to multiply each element of an array named A by the corresponding element of an array named B and store these results in an array named C, one need only type:

```
C←A×B
```

The equivalent operation in FORTRAN would have required the following steps:

```
      DIMENSION A(10), B(10), C(10)
      DO  1  I = 1, N
1     C(I) = A(I) * B(I)
```

Note also that in the FORTRAN program the parameter N must be previously specified. In the APL example, the user does not need to know how many elements are in the arrays A and B (although they must have the same length).

The power of APL can best be illustrated with some examples. In one case, an individual came to me late in the day with some circuit equations. For several days he had been trying to get a good run in a batch mode using FORTRAN; now, he needed the results the following day. He presented equations of the form

$$MX = Y$$

where M was a given six-by-six matrix, X was the unknown vector, and Y was six known numbers. He was reminded that we formally solve the above set of equations by writing:

$$X = M^{-1} Y$$

where M^{-1} is the inverse of the matrix M. The computer was then dialed up, and an APL function was loaded for inverting a matrix from a common library in the system. The following shows the output of the entire terminal session.

```
)1049273:ΛΔ
OPR:  APL WILL GO OFF THE AIR AT 1700......
011) 13.09.58 02/19/70 ALJONES

   A P L \ 3 6 0

      )LOAD 1 UTILITY
SAVED  16.38.15 01/28/70

      M←6 6⍴4.5 1.21 ¯.23 ¯.14 .76 .45 ....

      Y←19.7 3.45 2.1 ¯3.676 2.1 4.12

      X←⎕←(INV M)+.×Y

7.826908486  ¯11.13036082  4.421332176 ....

      M+.×X
19.7  3.45  2.1  ¯3.676  2.1  4.12

      )OFF HOLD
011  13.12.29 02/19/70 ALJ
CONNECTED      0.02.31  TO DATE     0.06.13
CPU TIME       0.00.05  TO DATE     0.00.10
```

The first statement after loading the UTILITY workspace says that we wish to store a six-by-six matrix under the name M. The elements of the matrix are strung out after the 'ρ' operator. The second line stores the values of Y, and the third line inverts the matrix M and performs the usual matrix product of that result with Y. The result was then stored in the variable X and printed out. As a check, we entered

```
M+.×X
```

and the values of the inputted vector Y were printed out.

The above example illustrates the powerful "desk calculator" mode. Note that very significant results were obtained without ever writing a program (although use was made of a common library function for inverting a matrix). To demonstrate the power and conciseness of the language, the function for inverting the matrix is listed:

```
    ∇ Z←INV M;I;J
[1]   M←⍉(1 0 +⍴M)⍴(,⍉M),~J←1<⍳I←1↑⍴M
[2]   →2×⍳0≠I←I-⍴⍴⍴M←1⌽(J,1)⌽[1] M-(J×M[;
      1])∘.×M[1;]←M[1;]÷M[1;1]
[3]   Z←M[;⍳1↑⍴M]
    ∇
```

Table I shows the monadic (one argument) and dyadic form (two arguments) of the primitive scalar functions in APL\360. This table is taken from the APL\360 User's Manual.[2] All of these functions can be used either in the desk calculator mode or in a function definition.

Monadic form f*B*		f	Dyadic form *A*f*B*	
Definition or example	Name		Name	Definition or example
+B ↔ 0+B	Plus	+	Plus	2+3.2 ↔ 5.2
-B ↔ 0-B	Negative	-	Minus	2-3.2 ↔ ¯1.2
×B ↔ (B>0)-(B<0)	Signum	×	Times	2×3.2 ↔ 6.4
÷B ↔ 1÷B	Reciprocal	÷	Divide	2÷3.2 ↔ 0.625
B ⌈B ⌊B 3.14 4 3 ¯3.14 ¯3 ¯4	Ceiling	⌈	Maximum	3⌈7 ↔ 7
	Floor	⌊	Minimum	3⌊7 ↔ 3
*B ↔ (2.71828..)*B	Exponential	*	Power	2*3 ↔ 8
⍟*N ↔ N ↔ *⍟N	Natural logarithm	⍟	Logarithm	A⍟B ↔ Log B base A A⍟B ↔ (⍟B)÷⍟A
|¯3.14 ↔ 3.14	Magnitude	|	Residue	Case A|B A≠0 B-(|A)×⌊B÷|A A=0,B≥0 B A=0,B<0 Domain error
!0 ↔ 1 !B ↔ B×!B-1 or !B ↔ Gamma(B+1)	Factorial	!	Binomial coefficient	A!B ↔ (!B)÷(!A)×!B-A 2!5 ↔ 10 3!5 ↔ 10
?B ↔ Random choice from ⍳B	Roll	?	Deal	A Mixed Function (See Table 3.8)
○B ↔ B×3.14159...	Pi times	○	Circular	See Table at left
~1 ↔ 0 ~0 ↔1	Not	~		
		∧	And	A B A∧B A∨B A⍲B A⍱B 0 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 1 1 1 1 0 0
		∨	Or	
		⍲	Nand	
		⍱	Nor	
		<	Less	Relations Result is 1 if the relation holds, 0 if it does not: 3≤7 ↔ 1 7≤3 ↔ 0
		≤	Not greater	
		=	Equal	
		≥	Not less	
		>	Greater	
		≠	Not Equal	

(-A)○B	A	A○B
(1-B*2)*.5	0	(1-B*2)*.5
Arcsin B	1	Sine B
Arccos B	2	Cosine B
Arctan B	3	Tangent B
(¯1+B*2)*.5	4	(1+B*2)*.5
Arcsinh B	5	Sinh B
Arccosh B	6	Cosh B
Arctanh B	7	Tanh B

Table of Dyadic ○ Functions

Table 1 — Primitive Scalar Functions

EXAMPLES OF THE USE OF *APL*\360

Solution of Differential Equations

One of our most common uses of *APL*\360 for analysis of mechanical and electrical systems is in the solution of a set of ordinary differential equations. Consider a simple example of the use of a fourth-order Runge-Kutta routine and the problem of a ball free-falling until it hits a spring-dashpot system. The equation of motion of the ball is:

$$\frac{d^2x}{dt^2} = -g \qquad x > 0$$

$$\frac{d^2x}{dt^2} = -g\left[1 + \left(kx + c\,\frac{dx}{dt}\right)\Big/W\right] \qquad x < 0$$

and the initial conditions are:

$$\text{at } t = 0 \quad x = h$$

$$\frac{dx}{dt} = 0$$

We have formulated the problem of a ball dropped initially from a height h. It strikes a foundation which is characterized by a spring-dashpot system with parameters k and c respectively. Following is a listing of the APL functions used to solve this problem:

```
      ∇BALL[⎕]∇
    ∇ Z←ΔT BALL TF
[1]   X←(Z←H),T←0
[2]   Z←Z,1↑X←T RUNGE X
[3]   →2×TF>T←T+ΔT
    ∇

      ∇RUNGE[⎕]∇
    ∇ Z←T RUNGE X;K
[1]   K1←ΔT×T EQN X
[2]   K2←ΔT×(T+0.5×ΔT) EQN X+0.5×K1
[3]   K3←ΔT×(T+0.5×ΔT) EQN X+0.5×K2
[4]   K4←ΔT×(T+ΔT) EQN X+K3
[5]   Z←X+(K1+K4+2×K2+K3)÷6
    ∇

      ∇EQN[⎕]∇
    ∇ Z←T EQN X
[1]   Z←X[2],-G×1+((X[1]<0)×(K̲×X[1])+C×X[
      2])÷W
    ∇

      H←50
      G←386
      K̲←.5
      W←1
      C←.005
      X←.05 BALL 4
      T←0,.05×⍳80
```

The first function, *BALL* , sets the time increment and final time. The function *RUNGE* defines the Runge-Kutta algorithm. Notice that the function *RUNGE* is not only very easy to write once the algorithm is known, but also that the algorithm is easily understood by reading the function. That is, the language is closer to mathematics than any other existing language. The function *RUNGE* solves a set of first-order differential equations, linear or nonlinear, homogeneous or nonhomogeneous. This function takes the values of the vector *X* (the dependent variables) at the value of *T* (the independent variable) and computes the values of *X* at *T* + *ΔT* . This is done according to the differential equations which are defined in the function EQN. If a change is made in the problem, it is necessary to change only the function EQN. Even the order of the system can be changed very simply. In this problem, the parameters *H*, *G*, *K̲*, *W* and *C* were set as "global" variables. This is far more convenient than having a program which requires one to read in a list of variables each time the program is executed. Definition as global variables allows any one of these to be simply changed before the next execution of the function. After the height of the ball as a function of time is computed, the results can be plotted by using a plotting function which is in the APL common library. To plot the response, one needs only give the command

```
      40  60  PLOT X VS T
```

The 40 and the 60 give the approximate size of the plot in lines and spaces respectively. The use of the *PLOT* function will be explained more fully later in this paper.

```
40 100 PLOT X VS T

 50oo
   | o
   |  o
   |
   |   o
 40|
   |    o
   |
   |
   |     o
 30|                      o
   |                    oo oo
   |      o
   |                   o     o
   |                  o       o
 20|       o
   |                 o
   |                           o           oooo
   |                                      o    o
   |                o           o               o
 10|        o                            o
   |                                             o
   |               o             o      o
   |                                              o
   |         o
  0|                                   o
   |              o               o                o
   |
   |                                  o
   |          o                    o                o
¯10|             o                   o               o
   |                                o
   |           oo
   |
   |
¯20|----|----|----|----|----|----|----|----|----|----|
 0.0       0.5       1.0       1.5       2.0       2.5
```

Curved Elastic Bar Analysis

We now consider the small, in-plane, deflections of a curved elastic bar. The configuration is shown in Figure 1. The load P is applied as shown. We wish to find the deflection of the loaded end in the direction of P and at right angles to P. We also wish to find the rotation of the section at the loaded end. This can be formulated using Castigliano's theorem[3].

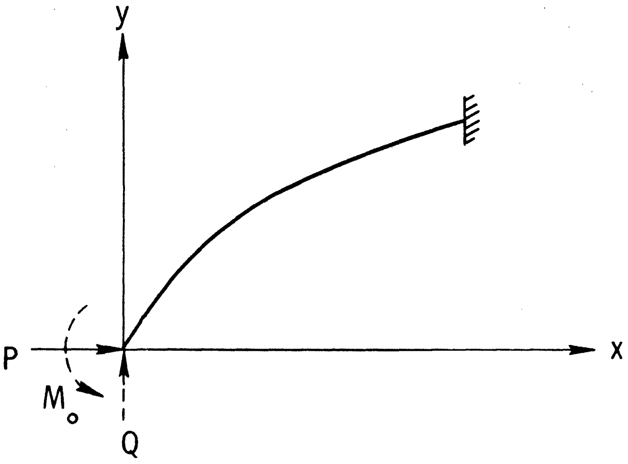


Figure 1 — Curved Elastic Bar Under Load

For thin bars, the strain energy U in the bar, as a function of the distance along the bar, is given by

$$U = \int_0^L (M^2(s)/2EI)\,ds$$

where the moment is given by

$$M = Py - Qx + M_o$$

where P is the applied load, Q is a dummy load, and M_o is a dummy moment. The desired deflections are obtained by differentiation and are given by:

$$\delta_P = (P/EI) \int_0^L y^2(s)\,ds$$

$$\delta_Q = (P/EI) \int_0^L x(s)\,y(s)\,ds$$

$$\psi = (P/EI) \int_0^L y\,ds$$

where δ_P is the deflection in the direction of P, δ_Q is the deflection perpendicular to the direction of P, and ψ is the angle of rotation. The bar can be broken up into straight line sections and the integrals in the above expressions carried out. The deflections can then be written,

$$\delta_P = (P/EI) \sum_{j=1}^{n-1} \Delta s_j\,(y_j^2 + y_j\,y_{j+1} + y_{j+1}^2)/3$$

$$\delta_Q = -(P/EI) \sum_{j=1}^{n-1} \Delta s_j\,(2x_j\,y_j + x_{j+1}\,y_j + x_j\,y_{j+1} + 2x_{j+1}\,y_{j+1})/6$$

$$\psi = (P/EI) \sum_{j=1}^{n-1} \Delta s_j\,(y_j + y_{j+1})/2$$

where Δs_j are the length of the sections given by

$$\Delta s_j = \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2}$$

and x_j, y_j are the coordinates of the points joining the sections. If the load P is applied in a direction which is not parallel to the x-axis but rather at an angle α, the above relationships can be used if the coordinate axes are rotated using the relationships

$$x = x' \cos\alpha + y' \sin\alpha$$

$$y = -x' \sin\alpha + y' \cos\alpha$$

where x', y' are the coordinates in the unrotated position. The APL function which computes ΔP, ΔQ, ψ is listed. The input consists of the angle α and x, y coordinates of the nodal points.

```
     ∇ K←ANGLE SPRING DATA;ΔS;M;X;Y
[1]    M← 2 2 ρM, ¯1 1 ×ΦM← 2 1 ○○ANGLE÷
       180
[2]    DATA←M+.×DATA
[3]    ΔS←¯1↓(+⌿(DATA-1ΦDATA)*2)*
       0.5
[4]    Y←DATA[2;]
[5]    X←DATA[1;]
[6]    K←÷ΔP←(÷3×E×I)×+/ΔS×¯1↓(Y*2)+(Y×1ΦY)
       +1ΦY*2
[7]    ΔQ←(÷6×E×I)×+/ΔS×¯1↓(2×X×Y)+(X×1ΦY)+
       (Y×1ΦX)+2×1ΦX←Y
[8]    PSI←(÷2×E×I)×+/ΔS×¯1↓Y+1ΦY
     ∇

       E←I←1
       CIR←2 1 ∘.○○,.05×ι10
       CIR[1;]←1-CIR[1;]

       0 SPRING CIR
1.28
       ΔP,ΔQ,PSI
0.7814  0.8521  0.9969
       ⍝ THE EXACT SOLUTION IS:
       ⍝ ΔP = .7854
       ⍝ ΔQ = .5
       ⍝ PSI = 1
```

The example here is that of a bar whose center line is a quarter of a circle. The exact solution is

$$\delta_P = \pi Pr^3/4EI$$

$$\delta_Q = Pr^3/2EI$$

$$\psi = Pr^2/EI$$

The quarter circle was broken into ten straight-line sections. It can be seen that the errors in δ_P, δ_Q, and ψ are 0.5%, 0.1% and 0.3% respectively.

Impact Printing

The above two problems were used as simple examples to show the ease of using APL. Of course, most problems are much more complex. One such problem is that of printing simulation. In modeling the dynamics of printing, the hammer was taken as a wave propagation medium, and the paper was characterized as a nonlinear dissipative medium. In addition, the dynamics of the type slug were taken into account. The entire system was programmed using APL. As the results were computed, they were plotted using the typewriter. This allowed us to monitor the effect of changing a given parameter and to further understand some of the experiments performed. With this flexible analysis engineers were able to interpret the shape of the force-time traces of the printing which were not previously understood.

Dynamic Buckling of Thin Rods

An engineer who is a specialist in engineering mechanics in our group solved another quite complex problem using APL; this problem involved the dynamic deflections of thin rods sliding in elastic guides. The rods were axially loaded, and the guides had the effect of preventing the buckling of the rods. The problem consisted of the dynamic coupling of the elastic deflections of both the rod and the guides through the everchanging contact points and contact forces between them. Programming in APL greatly simplified the need of constantly expanding, contracting, and relocating these contact points. In fact, it is only through the terminal time-sharing, the speed of APL operation, and the plot display, that one was able to understand the effects of various assumptions and parameters well enough to make intelligent, realistic choices.

Contact Stresses in Layered Elastic Solids

Many features of the APL language have been used to full advantage in computing the stresses in a layered elastic media based upon theoretical analysis along the line of the Hertzian contact theory. A high-order Gaussian quadrature scheme was considered necessary for efficient numerical computations of the stresses and displacements. The matrix operations in the APL language make the parallel computation of several integrals in the Gaussian quadrature scheme possible, so that all the stress and displacement components are obtained at the same time. In contact stress analysis, the surface pressure and contact area are usually not known beforehand. The stress analyst is usually given the data on the geometry of the bodies in contact and the force between them. In layered elastic solids, the simple relation derived by Hertz to find the surface pressure and contact area is no longer valid. The quick man/machine interaction provided by the APL terminal system allows one to converge toward the surface pressure and contact area from a knowledge of the contact force and geometry of the contacting bodies.

Holography

APL has also been used by our department in the analysis of problems in holography. One such problem which was particularly easy to solve involved phase holograms. The question involved the computation of the phase and amplitude which should exist on a hologram. Only the phase is used in the physical reconstruction. Some experimental work indicated that reasonably good reconstruction could be made with this procedure.

If we consider only Fourier holograms, the field in the focal plane of a lens with focal length f is given by

$$u(x') = \int v(x) \exp\left\{\frac{2\pi i x x'}{\lambda f}\right\} dx$$

where λ is the wavelength of light, $v(x)$ is the desired field to be reconstructed and x and x' are the coordinate in the image plane and hologram plane respectively. Let

$$u(x') = A(x') \exp[i\phi(x')]$$

If the amplitude information is discarded, the reconstructed field will be

$$v'(x) = \int \exp\left[i\phi(x') - \frac{2\pi i x x'}{\lambda f}\right] dx$$

For the case where $v(x)$ is a uniform line of point sources, we would like to find the effect of varying the phases of the point sources on the reconstruction. When $v(x)$ is a line of point sources with phases ϕ_n, the field $u(x')$ is

$$u(x') = \sum_{n=0}^{N-1} \exp\left[i\phi_n + \frac{2\pi i n \Delta x x'}{\lambda f}\right]$$

An APL function was written which evaluated $u(x')$ for a random set of ϕ_n using a fast Fourier transform function.

The phase of $u(x')$ was then computed using an arctan function, and then the expression for $v'(x)$ was evaluated by replacing the integral by a sum using the trapezoidal rule.

The functions and an example of their use are given. The resulting graph shows that upon reconstruction there can be a great variance between points which should be uniform.

```
     ∇ Z←KINOFORM IN
[1]    Z←(÷¯1↑ρIN)×FFT 2 1 ∘.○PHASE CONJ
       FFT CONJ IN
[2]    Z←+/Z*2
     ∇

     ∇ Z←PHASE X
[1]    Z←(RE X) ATAN IM X
     ∇

     ∇ Z←X ATAN Y
[1]    Z←(○(X<0)×¯1+2×Y≥0)+¯3○Y×(X≠0)×(X≠0)
       ÷X
[2]    Z←Z+○0.5×(X=0)××Y
     ∇

     ∇ Z←CONJ X
[1]    Z←(RE X) C-IM X
     ∇

     ∇ Z←SETUP M
[1]    Z← 2 1 ∘.○2×○
       1E¯6×?(ρM)ρ1000000
[2]    Z←(1 0 +ρZ)ρM,,Z
     ∇

     ∇ Z←N IMAGE IN
[1]    Z←(2,N)ρ0
[2]    Z[;(ι1)+N|IN[ι1;]]←IN[1 2 +ι1;]
     ∇

       IN←SETUP ¯3 ¯2 ¯1 1 2 4
       R←KINOFORM 16 IMAGE IN
       20 50 PLOT (8⌽R) VS ¯9+ι16
```

```
   20 50 PLOT (8⌽R) VS ¯9+ι16

0.20|
    |
    |               □
    |               □
    |               □     □
0.15|               □     □
    |               □     □
    |               □     □     □
    |               □ □   □     □
    |             □ □ □   □ □   □
0.10|             □ □ □   □ □   □
    |             □ □ □   □ □   □
    |             □ □ □   □ □   □
    |             □ □ □   □ □   □
    |             □ □ □   □ □   □
0.05|       □     □ □ □   □ □   □
    |       □     □ □ □   □ □   □
    |       □   □ □ □ □ □ □ □   □
    |       □   □ □ □ □ □ □ □ □ □     □
    |   □   □ □ □ □ □ □ □ □ □ □ □   □ □
0.00|----|□---|----|----|----|----|----□----|----|
  ¯10        ¯5         0         5        10
```

MATHEMATICAL AIDS

Hypergeometric Functions

The generalized hypergeometric function is defined as

$$ {}_pF_q\left(\begin{matrix} a_1, a_2, \ldots, a_p \\ b_1, b_2, \ldots, b_q \end{matrix}\middle|\, z\right) = {}_pF_q\ (a_{p_i}\ ; b_{q_i}\ ; z) $$

$$ = \sum_{k=0}^{\infty} \frac{(a_1)_k\ (a_2)_k\ \ldots\ (a_p)_k\ z^k}{(b_1)_k\ (b_2)_k\ \ldots\ (b_q)_k\ k!} $$

where

$$ (a)_0 = 1,\ (a)_k = a(a+1)\ldots(a+k-1) = \frac{\Gamma(a+k)}{\Gamma(a)} $$

We can write this as

$$ F = 1 + \frac{a_1 a_2 \ldots a_p\ z}{b_1 b_2 \ldots b_q\ 1}\left(1 + \frac{(a_1+1)(a_2+1)\ldots(a_p+1)z}{(b_1+1)(b_2+1)\ldots(b_q+1)2} \cdot (1 + \ldots.)\ldots\right) $$

The base function in APL is defined in such a way that this expression can be written as a one-line function in APL.

```
     ∇ Z←N HYPERG X;I
[1]    Z←(⌽X×(×/A̲∘.+I-1)÷I××/B̲∘.+¯1+I←ιN)⊥1
     ∇
```

In this function, *N* is the number of terms computed, and the variables *A̲* and *B̲* are the vectors which contain the parameters a_p, b_q. As a very simple special case,

$$ {}_0F_0\ (z) = e^z $$

This is computed in APL by setting `A̲←ι0,B̲←ι0`.

```
          A̲←B̲←ι0
          5 HYPERG 1
    2.708333333
          10 HYPERG 1
    2.718281526
          *1
    2.718281828
```

The Bessel function $J_0(x)$ can be computed by setting `A̲←ι0` and `B̲←,1` since

$$ J_0(x) = {}_0F_1\ (1; -1/4\, x^2) $$

```
          B̲←ι1
          10 HYPERG ¯.25×.5*2
    0.9384698072
          BJ0 .5
    0.9384698087
```

This was checked using an APL function which computes $J_o(x)$ by a rational approximation.

Fast Fourier Transform

The fast Fourier transform is a method of computing the series

$$x(j) = \sum_{k=0}^{N-1} A(k)\, e^{2\pi ijk/N}$$

with approximately $N \log_2 N$ computations instead of the usual N^2 computations. The algorithm was published by Cooley and Tukey[4] in 1965. However, the basic method was first derived in 1942 by Danielson and Lanczos[5]. The algorithm is based on the fact that, if N is highly composite (that is, has many factors), the sum can be rearranged to permit more efficient computation. The most advantageous case is when N is a power of 2, i.e., $N = 2^m$. In this case, the single sum over N terms is replaced by m sums, each one of which contains 2 terms. Space does not permit a complete explanation of the APL program. Below is an example of its use. A listing of the function appears in the Appendix.

```
      FFT 2 8ρ(8ρ1),8ρ0

8.000E0    ¯4.210E¯16 ¯3.488E¯16 ¯7.224E¯17
       0.000E0     7.224E¯17  3.488E¯16
       4.210E¯16
0.000E0     1.744E¯16  3.488E¯16  1.744E¯16
       6.976E¯16  1.744E¯16  3.488E¯16
       1.744E¯16
```

In this example, the function computed the transform of the input data which had all its real parts equal to one and its imaginary parts equal to zero. The exact result is for the first element to be 8 and all others zero. The second example takes the transform of an array of random integers and then inverts the result as a check.

```
    Z←FFT ⎕←X←?2 8ρ10

2    1    7    9    7    8    8   10
9    3    4    4    6    6    9    5
    Z

52           ¯1.414       ¯6            ¯2.929
      ¯4           1.414        ¯6
     ¯17.07
46           ¯5.071       ¯8            ¯0.2426
      10           9.071        12
       8.243

    .125×CONJ FFT CONJ Z

  2             1             7             9
        7             8             8
       10
  9             3             4             4
        6             6             9
        5
```

Numerical Quadrature

An APL function was written for the computation of the integral of a known function between the limits 'a' and 'b' using the 64 point Gaussian integration scheme. The two functions which accomplish this are *GINTEG* and *GSTEP* which are listed in the Appendix.

To use these functions, one merely enters the statement

```
          GINTEG FN GSTEP A,B,N
```

where *FN* is the name of the function to be integrated, *A* is the lower limit, *B* is the upper limit, and *N* is the number of subintervals to be used. For most cases, *N* is set equal to one. Following is an example of the use of this function to compute sin x from 0 to π and the function $y = |x|$ from -1 to 1.

```
          GINTEG 1○ GSTEP 0,○1
    1.9999999999999996

          GINTEG | GSTEP ¯1 1
    1.000197739390638

          GINTEG | GSTEP ¯1 1 2
    0.9999999999999979
```

The above functions could not be used for tabulated data unless one were willing to interpolate to find the value of the function at the necessary points. In the case of tabulated data, a trapezoidal function has been written.

```
      ∇ Z←TRAP XY;X;N
  [1]   Z←0.5×+/(N↓XY)×(1↓X,¯1↑X)-X[1],
         ¯1↓X←(N←0.5×ρXY)↑XY←,XY
      ∇
```

In this function, the data is stored in the vector X and Y. It is used by entering

```
      Y←0 1 2.8 3.1 3.5 2.9 2.1 1.1
      X←0 .9 1.6 2.3 3.1 3.9 4.5 5.4
      TRAP X,Y
11.985
```

Interpolation

Given an array y_j and an array x_j (x_j in order), we wish to find the points y' corresponding to x' where y' is obtained by linear interpolation,

$$y' = y_j + (x' - x_j)\,(y_{j+1} - y_j)/(x_{j+1} - x_j),$$

and j is determined by finding the interval where $x_j \leq x' < x_{j+1}$. This can be written as a one-line APL function, not only for the case of a single value of x'. The same function is used for the case when x' is an array of values for which we wish to find the corresponding array of y' values. The APL function is listed in the Appendix.

An example which follows illustrates its use in finding the values of sin x when several tabulated values are given

```
        Y←1○X←0 .1 .19 .27 .35 .39 .44 .48
        (X̲←.1 .2 .3 .4) LINT X,Y
 0.0998334  0.198593  0.295294  0.389339

        1○X̲
 0.0998334  0.198669  0.29552  0.389418
```

Statistical Programs

APL is particularly suited for writing statistical programs. This is due to its ability to handle arrays in a very concise manner. The usual statistical routine in FORTRAN has many DO-loops and DO-loops within DO-loops. Since there are so many possible statistical functions that can be written, we will give only a very simple example to illustrate the ease of using APL for statistics.

```
      ∇ STAT X
 [1]    'N = ';N←ρX
 [2]    'MEAN = ';M←(+/X)÷N
 [3]    'VARIANCE = ';VARIANCE←(+/(X-M)*
        2)÷N-1
 [4]    'STANDARD DEVIATION = ';SD←VARIANCE*
        0.5
 [5]    'MAX = ';⌈/X
 [6]    'MIN = ';⌊/X
      ∇
```

Following is a function for computing a number of random numbers taken from a normal distribution. It is based on the method of Box and Muller.[6]

```
      ∇ Z←NORMAL N;U
 [1]    U←1E¯6×¯0.5+?Nρ1000000
 [2]    Z←(Nρ(¯2×⍟(0.5×N)↑U)*
        0.5)×, 2 1 ∘.○2×(
        0.5×N)↓U
      ∇


        STAT NORMAL 100
 N = 100
 MEAN = 0.0457
 VARIANCE = 0.997
 STANDARD DEVIATION = 0.999
 MAX = 2.63
 MIN = ¯2.46
```

Complex Arithmetic

While APL does not have any built-in method for handling complex numbers, it is relatively easy to write functions for adding, subtracting, multiplying, and dividing complex numbers. A complex scalar is handled as a length two vector; a complex vector of length N is handled as a two by N matrix, etc. Functions for performing simple complex arithmetic are given in a report by A. D. Falkoff and K. E. Iverson.[7] A. D. Falkoff has also written functions for the computation of sine, cosine, exponential, etc. when the argument is complex. To add or subtract complex arrays, one merely uses regular addition and subtraction. The functions for complex multiplication and division are listed here.

```
      ∇ Z←A CMPY B
 [1]    Z←(ρA)ρ(,-⌿A×B),,+⌿A×⊖B
      ∇

      ∇ Z←A CDIV B
 [1]    Z←(A CMPY CONJ B)÷(ρA)ρ+⌿B*2
      ∇
```

Plotting Function

A function has been written in APL for the purpose of producing typewriter plots. This function was written by A. D. Falkoff. Examples of its use were given above. It allows one to plot several dependent variables against a single independent variable. The user specifies the approximate plot size, and the function determines the appropriate scale factors. To plot the variables *Y* and *Z* against *X*, the user enters:

```
      PLOTSIZE  PLOT  Y AND Z VS X
```

The functions *AND* and *VS* operate on the data and put it in the proper form for entry into the *PLOT* function. The variable *PLOTSIZE* is a length two vector which gives the plot size in lines and spaces. The writer has altered this function so that it allows one to rotate the plot by 90 degrees. It also enables one to plot several independent variables against the same number of dependent variables. Another feature allows the user to force consecutive plots to have the same scale factors. Examples of the use of these features are given below. Note that when coincidence of points occurs the typeball backspaces and overstrikes.

```
  10 30 PLOT X AND (⌽X) VS X←0,.1×⍳10

1.0*                   o
   | *               o
   |   *           o
   |     *       o
   |       *   o
0.5|         ⊛
   |       o   *
   |     o       *
   |   o           *
   | o               *
0.0⌽----|----|----|----*
 0.0       0.5       1.0
```

```
HS←1
PC←'□○⋆∘∇∆'
10 30 PLOT 0 1 3 4 6 4 7 5 3 0 0 2

10|
  |
  |
  |             □
  |         □   □
 5|         □   □ □
  |       □ □ □ □ □
  |     □ □ □ □ □ □ □
  |     □ □ □ □ □ □ □     □
  |   □ □ □ □ □ □ □ □     □
 0|-□--|----|----|----◫-□--|----|
  0         5        10        15
```

The second plot is a histogram. This is done by setting the global variable *HS* to a value of one. The plot characters were respecified before plotting. Also, notice that if the right-hand argument is a vector, this vector is plotted versus its index.

A "circle" can be plotted as follows:

```
HS←0
PC←'○⋆∘∇∆□'
10 30 PLOT ⍉2 1 ∘.○(10×⍳36)÷180

 1|      oo o oo
  |   oo         oo
  |oo               oo
  |o                 o
  o                   o
 0o                   o
  o                   o
  |o                 o
  | o               o
  |   oo         oo
¯1|----|-oo-ϕ-oo-|----|
 ¯1        0         1
```

A plot can be rotated by setting the global variable *ROT* equal to one.

```
ROT←1
X←0,.1×⍳20
20 30 PLOT (1○X) AND (2○X) VS X

  ¯1        0         1
0.0|----|----ϕ----|----*
   |            o      *
   |               o *
   |               * o
   |            *      o
0.5|         *         o
   |      *            o
   |   *             o
   | *             o
   *            o
1.0*         o
   *      o
   | * o
   | o *
   o      *
1.5o         *
   o            *
   | o             *
   |   o             *
   |      o            *
2.0|         o         *
```

A plot of more than one dependent variable against the same number of independent variables can be made by setting the global variable *MULT* equal to a vector. The elements of each indicate how many elements are in each variable. An example is given.

```
ROT←0
Y1←X1←0,⍳10
X2←0 2.1 3.9 5 6.8 8.8
Y2←2 1.5 4.5 5.8 7.8 7
MULT←11 6
10 40 PLOT (Y1,Y2) VS X1,X2

10|                    o
  |                 o
  |             * o
  |             o   *
  |         * o
 5|       * o
  |       o
  |     o
  *   ⊛
  | o
 0ϕ----|----|----|----|
  0         5        10
```

To return to a normal plot *MULT* is respecified to ⍳0. The scale factors from the previous plot are used if the global variable *SCALE* is set equal to one instead of its normal value of 0.

CONCLUSION

It is now evident that the claims made for the efficiency and adaptability of *APL*\360 are justified. The wide range of applications, coupled with the accessibility and time-saving capability of this system, promise even greater acceptance among our engineers.

ACKNOWLEDGMENTS

The author would like to thank Drs. W. T. Chen and J. K. Kuo for their help with sections of this paper. Also, Mr. A. D. Falkoff and Miss Colleen Conroy of IBM Research Center, Yorktown Heights, were of great assistance during the period of learning *APL*\360. Mr. W. H. Niehoff, Jr. was instrumental in getting the *APL*\360 System installed at Endicott. The author is immensely grateful to Dr. K. E. Iverson for creating the APL language and, with A. D. Falkoff, L. M. Breed and R. H. Lathwell, developing it into the superb system it is. Also noteworthy are the contributions of Dr. A. J. Lavin who introduced the department to TELTRAN, and S. W. Reynolds who introduced *APL*\360. Finally Dr. W. C. Rodgers and Mr. J. P. Kirk aided in the development of some of the functions listed in this paper.

REFERENCES

1. Iverson, K. E., A Programming Language (John Wiley and Sons, Inc., New York 1962).

2. Falkoff, A. D., and Iverson, K. E., *APL*\360 User's Manual (International Business Machines Corporation, 1968).

3. Timoshenko, S., Strength of Materials (D. van Nostrand Co., New York, 1955), pp. 378, 379.

4. Cooley, J. W., and Tukey, J. W., "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. of Comput., 19, 297-301 (1965).

5. Danielson, G. C., and Lanczos, C., "Some Improvements in Practical Fourier Analysis and Their Application to X-ray Scattering from Liquids," J. Franklin Institute 233, 365-380, 435-452 (1942).

6. Box, G. E. P., and Muller, M. E., "A Note on the Generation of Random Normal Deviates," Ann. Math. Stat. 38, 601-611 (1958).

7. Falkoff, A. D., and Iverson, K. E., "The *APL*\360 Terminal System," IBM Research Report RC 1922, October 1967.

APPENDIX

Below are listings of APL functions referred to in the text.

Fast Fourier Transform

```
      ∇ Z←FFT X;C;D;E;J;K;L;LL;M;N;O
[1]     LL←⌊2*-O-ιM←⌊2⍟N,0ρE←1-2×~O←ι1,J←ιL
        ←0,0ρK←ιN←¯1↑ρX
[2]     →(M>L←L+1)/1+ρρJ←J,Nρ 0 1 ∘.=(
        2*L)ρ1
[3]     Z←X[;(L←O)+(⌽LL)+.×J←(M,N)ρJ]
[4]     X← 2 1 ∘.○○(-O-K)÷¯1↑LL
[5]     Z←Z[;K-,LL[L]×J[L;]]+(ρZ)ρ(-⌿X[;D]×
        Z[;C]),+⌿X[;D←O+NρLL[E+M-L]×-O-ι
        2×LL[L]]×⊖Z[;C←K+,LL[L]×0=J[L;]]
[6]     →((M+O)>L←L+1)/5
      ∇
```

Linear Interpolation

```
      ∇ Z←X LINT V;M;P
[1]     Z←V[M+P]+(V[M+1+P]-V[M+P])×(X-V[P])
        ÷V[P+1]-V[P←1⌈(¯1+M)⌊+/X∘.≥V[ιM←
        0.5×ρV←,V]]
      ∇
```

Gaussian Integration

```
      ∇ QA←GSTEP X;A
[1]     A←X[1]+(QΔ←0.5÷X[3]÷X[2]-X[1])×¯1+
        2×ι(X←X,1)[3]
[2]     QA←(,A∘.+QA),,A∘.-QA←QΔ×
        0.9993050417357721 0.9963401167719553
        0.9910133714767443 0.9833362538846226
        0.973326827789911 0.9610087996520537
        0.9464113748584028 0.9295691721319396
        0.9105221370785028 0.8893154459951141
        0.8659993981540928 0.8406292962525803
        0.8132653151227976 0.7839723589433414
        0.7528199072605319 0.7198818501716108
        0.6852363130542332 0.6489654712546573
        0.6111553551723932 0.571895646202634
        0.5312794640198945 0.489403145707053
        0.4463660172534641 0.4022701579639916
        0.3572201583376681 0.3113228719902109
        0.2646871622087674 0.2174236437400071
        0.1696444204239928 0.1214628192961206
        0.0729931217877904 0.0243502926634244
      ∇

      ∇ Z←GINTEG X
[1]     Z←QΔ×(+/[ρZ]((¯1↓Z),2,
        0.5×¯1↑Z)ρ,X)+.×(0.5×¯1↑Z←(ι1=ρρX),ρX)ρ
        0.0017832807216964333
        0.004147033260562468
        0.006504457968978363
        0.008846759826363948
        0.01116813946013113
        0.01346304789671864
        0.01572603047602472
        0.01795171577569734
        0.02013482315353021
        0.02227017380838325
        0.02435270256871087
        0.02637746971505466
        0.02833967261425948
        0.03023465707240248
        0.03205792835485155
        0.03380516183714161
        0.03547221325688238
        0.03705512854024005
        0.03855015317861563
        0.03995374113272034
        0.04126256324262353
        0.04247351512365359
        0.04358372452932345
        0.04459055816375656
        0.04549162792741814
        0.04628479658131442
        0.04696818281621002
        0.04754016571483031
        0.04799938859645831
        0.04834476223480296
        0.04857546744150343
        0.04869095700913972
      ∇
```

AN EXPANDED TERMINAL DESIGN FOR CLINICAL MEDICAL RESEARCH*

H. William Perlis, Steven E. Wixson, Eugene M. Strand
University of Alabama Medical Center, Birmingham, Alabama

ABSTRACT

A terminal is described for bedside research and patient care. Unique features are 1) physiological instrumentation considered an integral part of the terminal and 2) a flexible terminally oriented program structure.

INTRODUCTION

The University of Alabama Myocardial Infarction Research Unit (MIRU) supports clinical research on patients who have sustained a myocardial infarction (heart attack). Patient rooms provide the environment for intensive coronary care and research. Two laboratories facilitate study of critically ill patients with complicating conditions, such as, shock, congestive heart failure and severe arrhythmias. The digital computer housed adjacent to the patient rooms is dedicated to on-line real-time MIRU research.

A remote terminal complex has been designed for medical research and patient care. The terminal, located at the patient's bedside, consists of modular physiological instrumentation with both digital and analog interface, a storage display scope and numeric/special function keyboard.

A sub-executive supports the terminal with core resident terminal routines. Terminal control blocks and patient control blocks provide asynchronous operation of devices.

TERMINAL HARDWARE

Other medical research terminals have been described that provide for digital data acquisition and display of alphanumeric and graphic information.[1-2] The MIRU terminal includes the conventional display and keyboard. A unique feature of the MIRU terminal is that the bedside physiological instrumentation and the transducers can be considered part of the terminal. Figure 1 is a schematic of the MIRU terminal system.

The laboratory terminals include a graphic display scope, numeric/special function keyboard, and 14 channels of analog instrumentation. Similar terminals in the 'uncomplicated patient' rooms include only 7 instrumentation channels. The nurses' station terminal has only the display scope and keyboard which can be switched to work for any bed. Figure 2 shows an overall view of the 'uncomplicated patient' terminal. See the Appendix for details of the computer and terminal hardware.

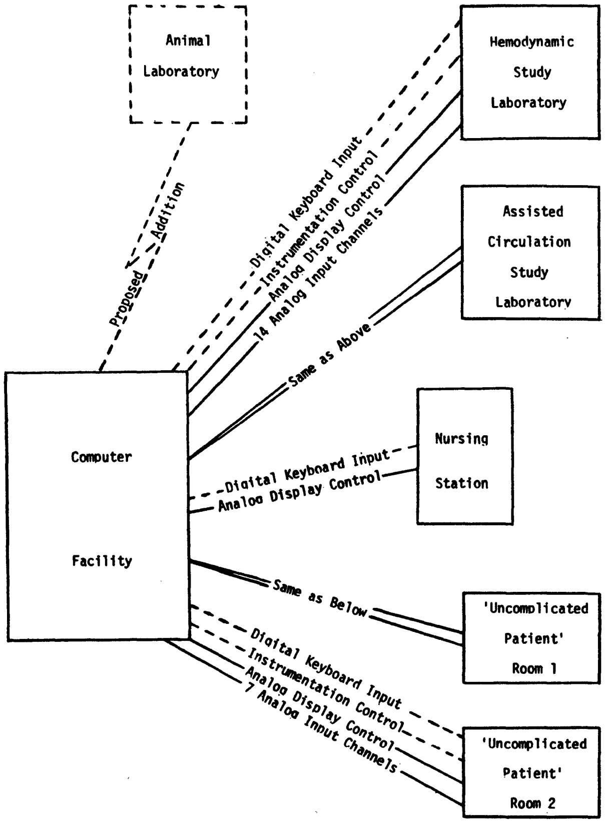

Animal Laboratory
Hemodynamic Study Laboratory
Proposed Addition
Digital Keyboard Input
Instrumentation Control
Analog Display Control
14 Analog Input Channels
Assisted Circulation Study Laboratory
Same as Above
Nursing Station
Computer Facility
Digital Keyboard Input
Analog Display Control
Same as Below
'Uncomplicated Patient' Room 1
Digital Keyboard Input
Instrumentation Control
Analog Display Control
7 Analog Input Channels
'Uncomplicated Patient' Room 2

Figure 1. Myocardial Infarction Research Unit

Display System

The display is based around a storage scope. The storage scope can display information on its face without external buffering and refreshing. A 'CRT' type of display was selected for the

---

*Supported In Part by U. S. Public Health Service Contract No. PH 43-67-1441

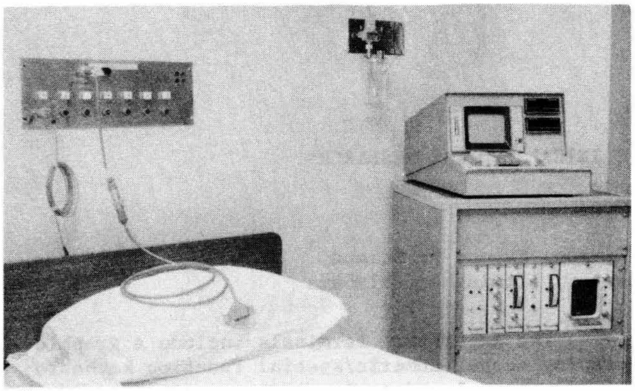


Figure 2. Overall View of MIRU Terminal

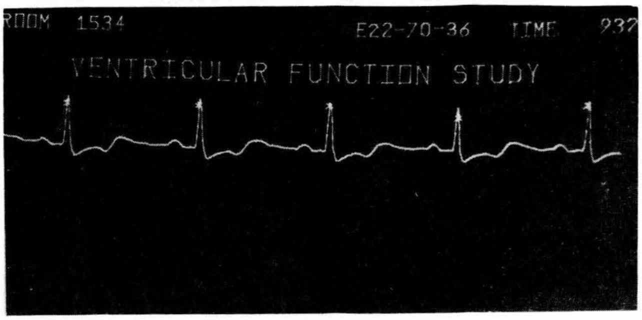


Figure 3. Graphic and Alphanumeric Display

bedside terminal because it provides alphanumerics, graphics, fast generation of information, and quietness. The storage display was selected because of its graphic capabilities and low cost. Figure 3 shows a display of digitized electrocardiographic data from a MIRU patient.

The display scope is essentially a high speed X-Y plotter. Its X and Y inputs are driven by two of the computer's digital to analog converters. The remote erase is wired to a computer 'electronic contact operate' point for program controlled erasing.

The storage display has limitations. However, these have not severely restricted our use of the display.

1) The basic display has a low contrast ratio which makes the display difficult to read in high light level areas. A television display is used for high contrast at some sacrifice in resolution.

2) There is no partial display erasing. An erase command erases the entire screen. An erase time of ½ second is required; software control permits overlap of the erase time and data processing.

3) There is no character and vector generation capabilities in the basic display scope. Several manufacturers provide hardware to perform these functions for a significant price increase. The MIRU display system uses software characters and vector generation. A conversion to hardware is not planned, since the computational demand is sufficiently small.

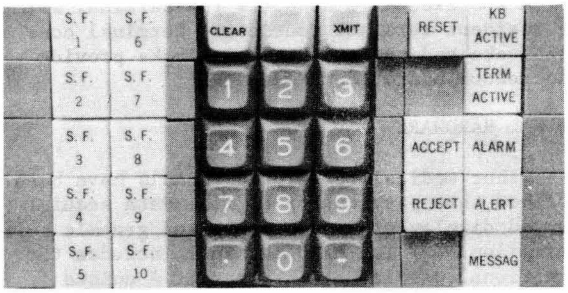
Keyboard

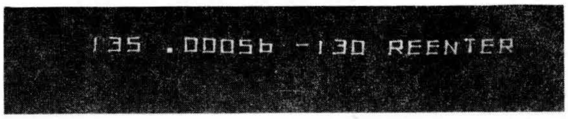
A basically numeric keyboard was designed since the input requirements show a preponderance of numeric and numerically codable information. An alphanumeric terminal is significantly more flexible but is therefore more difficult for nursing and medical technicians to operate.

The keyboard, shown in Figure 4, serves as the control point for information entry and retrieval. The terminal can be used for data entry and program call up. The keyboard has numeric, terminal status, and special function sections. Keyboard keys are coded with an 8-bit code which permits extension to a full EBCDIC, or ASCII keyboard if needed. The keys and lights are connected to the computer's digital input and output points. Depression of a key causes the computer to be interrupted. The interrupt is serviced by a terminal input/output control program.

The numeric section contains the data entry buttons: Numbers 0-9, minus sign, space or blank, and decimal point. An important feature of keyboard data entry is that these characters are displayed on the top line of the display scope for visual verification as each key is depressed. (See Figure 5) The characters displayed on the scope are buffered in the computer. Software restricts to fifteen the number of characters entered.

Terminal status is shown by lights and audio tones. The keyboard active light is turned





135 .00056 -130 REENTER

Figure 5. Visual Verification of Keyed Data

on when a data entry button is pressed. This shows the presence of data in the keyboard buffer. It is turned off when the data is transferred to a program, the clear button is depressed, or the input buffer is overrun.

A terminal active light is turned on when a terminal is dedicated to a program. This shows that a program expects to receive data from or display information at that terminal. The light is turned off when the program releases control of the terminal.

Alarm, alert, and message lights are turned on by programs to attract attention. These show detection by the program of critical conditions or the desire of the program to express itself. Functionally all lights are the same with the only difference being the audio tone sounded.

The special function section has action buttons and program call-up buttons. The action buttons are XMIT, CLEAR, RESET, ACCEPT, REJECT, and the ALARM, ALERT and MESSAGE RESPONSE.

The XMIT button signals the computer to call or recall a program. The CLEAR button is used to 'wipe out' data entries from the keyboard buffer. The ACCEPT and REJECT buttons set codes into the data buffer area and then recall a program. These are used in verification of important data. The RESET button resets the terminal, clears the keyboard buffer, and recalls the program which had control of the terminal. The ALARM, ALERT, and MESSAGE buttons recall the programs that turned on the respective lights.

The special function buttons labelled S.F.1 through S.F.10 make the terminal easier to use. These buttons call up the 10 programs most frequently used at that terminal. Alternatively, the nurse calls up a program by entering a three digit name and depressing the XMIT key.

Physiological Instrumentation

The physiological instrumentation, shown in Figure 6, provides flexible, easy to use, research quality analog signals. The signals are for monitoring in the patient room or nurses' station and for input to the computer.

A key feature of the instrumentation is modularity. In patient rooms, instrument cabinets accept up to 7 channels of instrumentation. In the laboratories, 14 channels of instrumentation are available plus an 8 channel monitor and oscillographic recorder. This allows any combination of instruments to be assembled for research or clinical care.

A computer interface provides a powerful two way digital information transfer between the computer and any instrument and one way transfer from instrument transducer connector to computer. Instruments send module type and module status information. Instruments receive polling signals and function commands. Transducers send transducer type information.

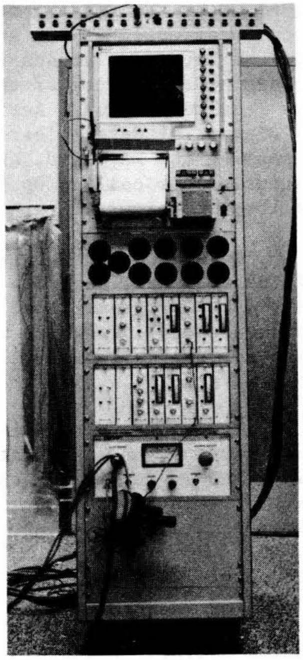


Figure 6. Physiological Instrumentation

Transducer panels are located above patient bed's for connecting instrument transducers to modules. There is one connector for each module position in the cabinet. Above each connector is the 'transducer active' button used to signal the computer. These buttons are essentially part of the keyboard but are located on the transducer panel to simplify operation.

When an analog signal is needed, the nurse connects the transducer, plugs in a signal conditioning module and presses the 'transducer active' button. The computer reads digital information from the conditioning module and transducer connector. A computer program checks codes to insure proper setup of the instrumentation, and using the display and keyboard guides the calibration of the module-transducer combination. A light in the 'transducer active' button signifies the signal is on-line.

A second pressing of the transducer active button recalls the program to recalibrate or to take the signal off-line. An off-line request removes calibration tables and turns off the transducer active light.

TERMINAL SOFTWARE

The specifications for the software terminal support included fast response to requests, software vector and character generation, ease of programming, variable function in the physiological instrumentation, and flexibility of data entry.

Implementation of these specifications led to a system which includes:

1) An in-core terminal input-output control routine (TIOCR),

2) A core partition reserved primarily for programs receiving data from keyboards,

3) A unique FORTRAN programming structure, and

4) A system of linked and interrelated master control blocks.

TIOCR resides in core storage since each keyboard key depression causes the computer to be interrupted. The patience of nursing personnel and the computational power of the computer would be taxed by the loading of this routine to service interrupts. TIOCR services the interrupts for the keyboard, senses the code of the key to determine the nature of the request, and performs the required action. Actions include buffering an element of data, clearing the keyboard buffer, resetting the keyboard, queuing a program for execution, or calling up the instrumentation calibration program.

To enhance the conversational response time of the terminal, one core partition has been reserved for programs whose maximum execution time is one second. Programs working conversationally with nursing personnel will usually execute in this 4K area.

The programming structure, known as the task concept, and the master control blocks are discussed in detail. These enhance the MIRU system beyond what was needed to satisfy the terminal requirements.

Task Concept

The basic unit of work in the MIRU system is the 'task'. A task is defined as a disk loadable program that is active for a specific patient and is uniquely identified by a program name and a bed number. A task may be of a few hundred milliseconds duration, (e.g., a summary display); or it may be for the duration of the patient's stay (e.g., a monitoring program executed at periodic intervals). Tasks can be initiated from the remote terminals or through other tasks.

The task concept dynamically provides system facilities to each patient in the research unit. The bedside terminal will be dedicated to any task for entry or display of information. Core and disk working storage are dynamically allocated for intermediate storage of parameters. Lights and audio alarms are available for signalling alarm and message conditions. Measurements such as heart rate, systolic or diastolic pressures are available in real-time to all programs, and any program can log information to the patient's data file.

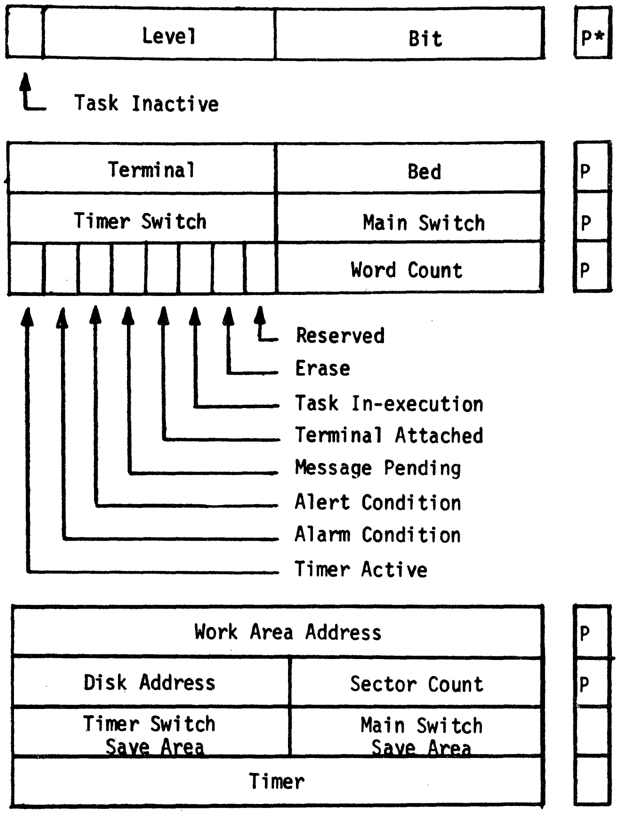
A program timer is provided for each task. The 1-second time base timer can be used to recall a program at a set interval, check data entry or response to alarm conditions, or schedule a program execution on a periodic basis.

When a program is initially requested, the TIOCR allocates eight words of core called a task block and initializes task parameters. The parameters include the program name, the terminal number, the bed number, two program switches, eight program status flags, the word count and address of in-core storage, the sector address and count of disk storage, a program switch save area, and a task timer. (See Figure 7. Task Block)

Program Organization

A task program is many program segments connected together under the control of two program sequence switches, the main and timer switches. The switch mechanism is the FORTRAN computed-go-to statement.[3] A program segment begins at one of the statement numbers in the computed-go-to statement and ends with a call to the MIRU executive exit routine. This gives a FORTRAN program multiple entry points. Thus, without being rolled-out (saved) onto disk, a program can be exited and entered at a later date. In-core storage is dynamically available to each program for the preservation of intermediate results. While a program is not active (executing), its partition is used by other programs. On recall for execution, the program is loaded from disk, and execution continues from the designated entry point.

This programming system conserves the overhead time that would otherwise be required to save an interrupted program on disk and frees a partition while a program is inactive. While one task waits for the entry of data from a keyboard, another task program is executing. The two program sequence switches provide the program with independent control paths for each task.



*NOTE: "P" means storage protected, and blank space means not protected.

Figure 7. Task Block

The first or main switch is the execution point following data entry via a remote terminal or response to an alarm, alert, or message signal. The second or timer switch is the execution point following a time-out of the task timer.

Since the two sequence switches are independent, each may point to a different segment of the program. A program requesting data can regain control after a specified time interval, whether or not data has been entered.

Task Entry

The TIOCR sets the switches to entry point one at task setup time. A task program begins with a CALL INITL, (i.e., the task initialization routine). Through this routine the program identifies the task being processed. The initialization routine designates the task as active and returns the sequence switches to the program.

A computed-go-to statement follows the CALL INITL.

```
GO TO (100,200,300,400,500,...), IPRSW
```

The numbers are the entry points to the program and 'IPRSW' is the main program switch. By selectively setting'IPRSW' as 1,2,...,N, the program can begin executing at entry points 100,200,..., respectively. This provides a powerful but easy mechanism to select the same or different entry point each time the program is executed.

Entry points 1-4 are reserved for system functions.

1) The task has been initialized through a keyboard request.

2) The task has been initialized through a program call from another task.

3) The task has been re-entered due to an external condition. For example, a system restart may have occurred, or the terminal reset key may have been depressed. This interrupts the normal execution flow of the task program.

4) The task has been re-started due to an execution error condition. Control is returned to the task program since it knows the significance of the error. Corrective action, such as shutting down a process, may be necessary before the task is exited.

Task Exits

The last executable statement of every program segment is a 'CALL MEXIT' (i.e., MIRU exit routine). Through this routine, a program communicates with the task block. The user can specify why he is exiting the program (exit type), where he wishes to return (main switch), where to return on a timer recall (timer switch), and the increment of time to remain inactive. The TIOCR saves this information in the task block, designates the task as inactive, and exits the program.

The TIOCR provides seven exit types for a task.

0) The task has completed all execution. The task block, work storage and terminal are released.

1-3) An alarm, alert or message condition is to be signalled. The respective light is turned on at the bedside and nurses' terminals. The task program remain inactive until a nurse depresses the respective respond key.

4) The task is to remain inactive for a specified interval and queued for execution when the interval is timed-out.

5-6) The task requests data entry from the terminal. The task remains inactive until the data entry key is depressed. Exit six erases the display at the time of data entry.

Master Control Blocks

All work within the MIRU system is associated with a specific patient. The terminal can do no work apart from a patient. Conceptually, the patient is represented in the computer by a table known as the bed control block (BCB). One BCB per patient is permanently stored in core.

The bed control block contains identification information for each patient. This forms a software link between the patient and the programs that are active for a patient.

Each program is called for a specific patient. Thus, the table set up to control a unit of work, the task block, links directly to a single BCB.

Two computer tables represent the terminal: 1) A terminal control block (TCB) associated with the keyboard and display portion of the terminal, and 2) A parameter control block (PCB) associated with the bedside physiological instrumentation.

The first, a fixed size control block for each terminal, is core resident. Basically, this block extends the BCB in that bedside terminals link one to one with the patient.

Structurally separating the TCB from the BCB permits independent and asynchronous function of multiple terminals for a single bed. Thus, the nurses' terminal and bedside terminal can simultaneously work for one bed.

The TCB contains information pertinent to the generation of displays and keying of data. Software support to the terminals provides alphanumeric and graphic display of information. Full

TABLE I
Terminal Control Block (TCB)

Words	P*	Description
1	P	Terminal number.
2	P	Address of task block in control of the terminal
3	P	Bed control block number.
4	P	Hardware bit mask for lights and display.
5-8		Display 'mode' scale factors.
9-10		Display origin for keyboard entry.
11-13		Display origin for FORTRAN IOCR.
14-21		Display scale factors.
22-31	P	Special function program names.
32-47		Keyboard buffer and pointer.
48		Time of data entry.
49	P	Digital input address.
50		Reserved.

*NOTE: 'P' means storage protected, and a blank space means not protected.

FORTRAN formatting capability is available for generating alphanumeric displays. Vector generation subroutines display graphic information. Vector and character generation routines reference the TCB for scaling factors. (See Table I. Terminal Control Block)

FORTRAN integer or real variables can be entered for a task program through the terminal keyboard. When a keyboard has been dedicated to a task, data can be entered for that task program. Up to 15 characters of information (numerics, minus sign, blank, decimal point) can be buffered in the TCB. Individual data words are separated by the blank, so more than a single value can be entered. For example, the string [130 .0005 -130] represents three distinct data entries. TIOCR subroutines move the character string from the TCB to the task program and convert it to FORTRAN real or integer values.

The TCB and a specific task block can be linked to work for a patient. The terminal display scope and keyboard can be dedicated to a task through a call to a TIOCR subroutine ATACH. This subroutine sets the terminal attached flag in the task block and places the task block address in the TCB.

Thus, a task program can generate a display or receive data from the keyboard without interference from another task. When a terminal is attached to a task, all other programs are prohibited by a software mask from using that terminal.

The second, a variable sized parameter control block, is dynamically reserved as an instrumentation module is placed on-line. This block contains addressing and scaling information unique to the module. Dynamic allocation of PCB's provides variable function in the bedside instrumentation.

Basically, the PCB's are variable extensions of the BCB, as an instrumentation module relates directly to one patient.

Within each BCB an area is reserved for the link to control blocks for active instrumentation modules. Two words are reserved for each 'transducer active' button in a patient room. Selecting a 'transducer active' button places a module on-line. When an instrumentation module is placed on-line, a parameter control block is dynamically reserved. The type and address of the PCB is placed in the BCB. (See Table II. BCB's and PCB's)

The TIOCR initializes the PCB with the module's analog multiplexor address and the calibration factors. Programs using the signal from a module can save physiological results in a variable size section of the PCB. The physiological results can be referenced by any program active for the PCB's associated bed.

Through the PCB, an instrumentation module can be related directly to a single task block. This TIOCR can either check that a module is not needed before placing it off-line or inform the task program that its signal is no longer available.

TABLE II
Bed and Parameter Control Blocks

Bed Control Block (BCB)

Words	P*	Description
0	P	BCB length.
1	P	Bed control block number.
2-5	P	Patient number.
6-12	P	Patient name.
13	P	Digital input/output addresses.
14-40		Physiological data storage.
41	P	Maximum number of PCB's.
42-**		Parameter block addresses.

Parameter Control Block (PCB)

Words	P*	Description
0	P	PCB length.
1	P	Multiplexor address of module.
2	P	Address of task block in control of module.
3	P	Slope calibration.
4	P	Zero calibration.
5+		Derived physiological parameters.

*NOTE: 'P' means storage protected, and a blank space means not protected.

**NOTE: =73 for laboratoreies
=47 for uncomplicated patient rooms.

CONCLUSION

The computer system has been operational since July, 1969, with a prototype of the terminal. The Myocardial Infarction Research Unit opened in October, 1969, with the computer and instrumentation system being used for research studies from two patient rooms and two laboratories. Instrumentation of an animal laboratory is projected for 1970.

Initial acceptance of the complex MIRU system by the clinical staff has been excellent. We attribute this acceptance to several steps taken during the development of the system.

1) The clinical staff was involved in the system planning from the beginning of the program. Their opinions concerning hardware functions and physical configurations were seriously considered and implemented whenever possible.

2) Prototypes were constructed for actual test in a clinical environment. This served to both test the equipment and to familiarize the staff with the system as it evolved.

3) The adoption of the 'expanded terminal' design minimized the number of unfamiliar tasks for an already overworked staff.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the efforts of Mr. Jeary Vogt, now of Shared Medical Systems Corp., Philadelphia, Pa., in the selection of monitoring equipment and for the concept and design of the digital computer interface for the bedside instrumentation.

APPENDIX

The medical research terminal has been developed within the environment of the IBM 1800 Data Acquisition and Control System[4] with the Multiprogramming Executive (MPX)[5].

The pertinent features of the IBM 1800 are: a) 32K words of 2 microsecond core, 16 bit word, 12 data channels, b) 3 direct access disk drives, 500K words per disk, 35K words per second transfer rate per disk, c) two 60K byte per second digital tape drives, d) a 300 increment per second flat bed digital plotter, e) card reader punch, 300 cards per minute read, 80 columns per second punch, f) two 20K samples per second analog to digital converters, program control resolution 8, 11, 14 bits and sign, 64 analog input points per converter, ±5 VOLTS, single ended, g) four digital to analog converters, wired to output voltages simultaneously h) 128 digital input, contact i) 128 electronic contact operate j) 32 process interrupt, contact, k) 16 process interrupt, voltages, l) 12 interrupt levels, m) 3 interval timers, 8 KHZ, 1 KHZ, 64 ms.

The graphic system is based around the Tektronix display scope. In laboratories where light levels are high, the Model 4501 Scan Converter is used to drive a Conrac television monitor. In the patient rooms where terminal activity is less, the Model 601 display scope (6" screen) is used. The Model 611 display scope (11" screen) is used at the nurses' station and on a portable terminal normally located in the doctor's lounge or computer room. The Model 611 gives the highest quality displays. The X inputs and Y input from all displays are connected in parallel respectively to two digital to analog converters. Each Z axis (beam control) is connected to a separate electronic contact operate point. The erase interval line from each display is connected to a digital input, contact point (transistor interface required). To accept the high speed computer-generated information, the Model 4501 requires a transistor switch from the Z axis signal to input 'write mode'.

The physiological instrumentation was assembled from standard medical sensors, Statham SM series instrumentation and locally built circuits. The back panel of the Statham cabinets and all instrument modules were rewired so that the analog output signal is on a common channel in all modules. Connector positions were thereby freed for the digital interface. Line drivers (operational amplifiers) were added to provide low output impedance for sending signals to multiple points. All transducer connectors were changed to 12 pin bayonet connectors. The extra pins are used for digital information.

The remote keyboard consists of 33 microswitch buttons and lights (Honeywell KB series). The data lines from each keyboard are connected parallel to 8 digital input contact points and the strobe to process input contact. Diode coding is used to generate data signals from the lighted push buttons. A high process interrupt level is used to insure millisecond response to a keystroke.

The IBM 1800 Multiprogramming Executive operating system is the real-time monitor for the computer. MPX provides automatic handling of interrupts from data input-output devices and user sources, automatic program scheduling, online hardware diagnostics, and timesharing for real-time routines, process programs, and background processing. The MIRU system provides these areas for program execution: Special coreload area (SPAR), coreload area, and variable core. These areas service, respectively, programs of high response and short execution (1 millisecond), medium response and medium execution (1 second), and slow response with variable execution.

REFERENCES

1) T. Allan Pryor, Reem M. Gardner, W. Clinton Day, "Computer System For Research and Clinical Application to Medicine", Digest of IEEE Computer Group Conference, June, 1969.

2) John J. Osborn, James O. Beaumont, John C. A. Raison, Jerome Russell, Frank Gerbode, "Measurement and Monitoring of Acutely Ill Patients by Digital Computer", Surgery 64:6, 1968.

3) IBM 1130-1800 Basic FORTRAN IV Language, IBM Systems Reference Library, Form C26-3715.

4) IBM 1800 Data Acquisition and Control System: Functional Characteristics, IBM System Reference Library, Form A26-5918.

5) IBM 1800 Multiprogramming Executive Operating System: Programmers Guide, IBM Systems Reference Library, Form C26-3720.

RELIABILITY OF INTEGRATED CIRCUITS

R. J. Sahni, Technology Division, Cogar Corporation

ABSTRACT

Reliability of integrated circuits has played the most significant role in winning them the front position in the electronics industry. The life of these circuits is several orders of magnitude larger than their predecessors'. Mean time between failures for a semiconductor chip with hundreds or even thousands of components is in millions of hours. It is only because of this phenomenal reliability that the assembly of very large and complex systems has become a reality.

To describe and evaluate the reliability of integrated circuits, new approaches have been sought for. This paper is a brief survey of such techniques.

I. INTRODUCTION

In this paper the reliability of a module with semiconductor LSI chip is discussed. The reliability of such a module can be defined as a two-fold assurance;

1. An assurance of protection from surroundings.
2. An assurance of survival while in use.

In the structure under consideration, the protection from impurities for silicon junctions is provided by phosphorous glass. Protection for aluminum interconnections from oxygen, moisture, and impurities in the atmosphere is provided by sputtered quartz, which acts as a good hermatic seal. The semiconductor chip is joined to a ceramic substrate by solder-reflow flip-chip bonding. (Other techniques like beam leads, spider bonding, etc. are not discussed here.) The bonds are protected from the ambient by a thick jelly-like sylgard.

While the modules are in use, they are subjected to various stresses. Failure modes introduced as a result of these stresses are discussed in the next section.

II. MODES OF FAILURES

A. Failures in Aluminum Interconnections on the Chip:

These failures are due to electromigration, which results from a combination of two 'machine-use' conditions:

(i) large currents flowing in various parts of the inter-connecting metal; and

(ii) elevated junction temperature due to large power dissipation in the chip.

Failures in aluminum inter-connections are of three types:

1. Aluminum land failures.
2. Failures of aluminum-silicon contacts.
3. 'Terminal failures', which are the failures of aluminum under a lead-tin pads.

1. Aluminum Land Failures:

The mode of failure of an aluminum land is an electrical open, which typically looks as shown in Figure 1. Such failures generally occur at the following places.

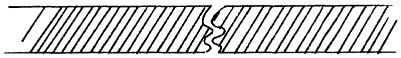


FIGURE 1

(a) Where the lands are narrower and/or thinner than the corresponding nominal values.

Thinning of the lands occurs because of variation in the thickness of the deposited film and the narrowing occurs mainly because of over-etching. It is evident that to maintain these variations to a minimum level is an essential step towards attaining good reliability. The spots where these two variations coincide constitute the potential sites for failures.

(b) Failures have also been seen to occur at places where a land passes over an oxide step. The presence of oxide steps on the surface of a processed wafer are an unavoidable outcome of the planar technology. The steepest steps occur at the places where the oxide is removed to make aluminum contacts to silicon. They range from 3000A° to 7000A°.

A Micro crack can develop on such steps because of preferential build up of aluminum atoms away from the straight edge during evaporation.(1) This can manifest itself as an electrical open or as a crack with fine hairline connections in between. The former would be weeded out in final test and thus be counted as a yield loss. The latter, however, would pass the final test but would result in a serious early failure in the field.

(c) Non-uniformity of grain size in aluminum film can also be a cause of stripe failures. Cracks have been observed at the boundaries where small grains meet large ones.(2) The movement of material and, therefore, the failure rate is directly proportional to the gradient in the composition of the film.(3)

(d) Temperature gradients along the stripes also enhance metal migration.(3) Such a gradient generally develops when a stripe passes over a highly powered resistor, which acts as a heat source.

2. Failure of Aluminum - Silicon Contacts:

As will be discussed in the next section, the mass transport of metal takes place in the direction of the electron flow.

Therefore, for resistors carrying large currents, there is a build up of aluminum at the negative terminal and depletion at the positive end. Referring to Figure 2, the arrows show the direction of the flow of the electrons in the resistor. When electromigration sets in near the negative end of the resistor, the aluminum atoms are carried in the direction of the arrows until they hit the aluminum - silicon interface. From that point they cannot go any further and therefore build up of aluminum results.

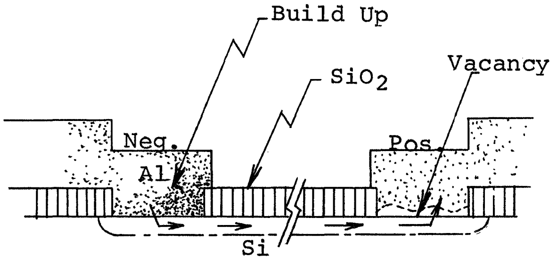


FIGURE 2

However, on the positive end, electromigration carries aluminum away from the contact. Since there is no aluminum below the contact to replace the migrated aluminum, a vacancy develops which, with time, spreads to an electrical open.

3. Terminal Failure:

Failures in the aluminum metallurgy under the lead-tin pads in solder reflow flip-chip bonding are called "terminal failures."

Figure 3(a) is a top view of such a failure. The shaded area is aluminum under the pad. Via hole in the glass is shown by dotted line. Dash-dotted circle represents the outer boundary of the reflowed lead-tin pad. The cross section of the pad is shown in Figure 3(b). As shown in the figure, migration of aluminum takes place all around the via hole edge. It is more clear from Figure 3(b) that such a migration disconnects the lead-tin pad from the aluminum land under it. The migration starts at a point of maximum current density and develops rapidly to the neighboring areas. This spreading continues until aluminum from all the periphery of the via hole has migrated.

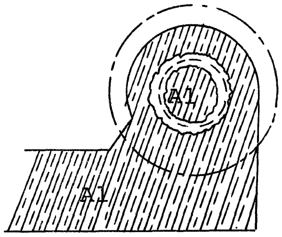
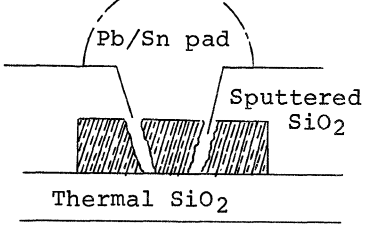


FIGURE 3(a)

FIGURE 3(b)



Since the migration takes place in the direction of the electron flow, only those pads have been observed to fail, where electrons travel from the pad to the aluminum land on the chip. In other words a positive terminal has never been observed to fail.

B. Failure of Chip-to-Substrate Pb-Sn Bonds:

The construction of solder reflow chip-to-substrate lead-tin joints can be described in terms of three essential steps: (4-6)

(a) Deposit a triple-layer of chromium, copper and gold over aluminum in via holes in sputtered quartz for outside contacts. Evaporate 5% tin, 95% lead solder on these sites. Reflow the pad.

(b) The fingers on the substrate are tinned with 10% Sn-90% Pb solder over silver-palladium paste.

(c) The chip is flipped over the ceramic substrate and the joint is made by reflowing.

The technique of obtaining these joints is called "controlled collapse" chip joining.(4) In this method the solderable area of the substrate land is limited by a glass dam and that of chip contact by the ball-limiting-metallurgy (BLM) of Cr-Cu-Au. This localization is essential, so that the surface tension in the molten pad and land solder support the chip until the joint solidifies.

The dominant mode of failure of such a bond is a crack near the top or bottom of the joint.(5) Thermal fatigue from turning on and off of the machine is the cause of the failure.

Corrosion can also be a problem if the joint is not well protected from the atmosphere.

III. PHYSICS OF FAILURE

A.. Electromigration

The mass motion that proceeds as a result of an applied electric field is known as electromigration. This phenomenon of current-induced mass transport has been studied extensively (7-10) by observing the motion of transverse scratches on the surface of gold, copper, and aluminum wires carrying large current densities (10^4A/cm^2). In the case of aluminum carrying a current density of about 10^4A/cm^2 the markers were observed to move as much as 50 /day and always towards the cathode.(9) Since the markers behave as vacancies, their movement towards the cathode is clear evidence of mass transport towards the anode, i.e., in the direction of the flow of electrons.

A similar physical phenomenon takes place in the case of metal films. In the structure of a metal film, the atoms which lie along the grain boundary, or on a defect, are loosely bound to the lattice compared to other atoms. Under the influence of external energy, such as heat, these atoms can be easily lifted from the lattice. In conductors carrying large current densities, such positive metal ions are under the influence of two opposing forces:

(a) Electronstatic force due to the applied field which is in the direction opposite to electron flow.

(b) A force due to the transfer of momentum from scattered electrons. This "electron friction" force tends to push the ions in the direction of electron flow.

The ions would drift in the direction of the predominating force. In the case of semiconductor devices, where aluminum lands carry high current densities (about 10^5A/cm^2) and are operated at temperatures of 50 to 100°C, aluminum ions pile up at the positive end, leaving vacancies at the negative end of the conductor. This observation suggests the dominance of the second force in aluminum films commonly used in semiconductor devices.

Without going further one can immediately spotlight a very fruitful challenge for semiconductor industry; namely, to develop a film in which the above mentioned two opposing forces precisely balance each other.

Theory:

For a manufacturer as well as a customer of LSI chips, the most relevant measure of reliability is the failure rate (F.R.) of the product. The failure rate is a reciprocal of mean time between failures (MTBF), where MTBF is an average of the times between any two consecutive failures.

The aim is to derive a mathematical expression for MTBF based on the physcis of the problem. The author has chosen not to present the derivation of the expression in this report. Those interested in detailed steps and accompanying assumptions may refer to the literature.(11, 12)

The dependence of the failure rate on various physical parameters is:

(a) Failure rate is directly proportional to temperature (T), because the higher the thermal energy available the easier it will be for the ions to lift themselves up from the potential well of the lattice.

(b) Failure rate is also directly proportional to current density (J) because the stronger the "electron wind" the faster the activated ions will be carried away and the quicker the failure will result.

(c) It is inversely proportional to the activation energy (ΔH), because the higher the activation energy the more difficult it is to lift an ion and create a vacancy. (Activation energy, ΔH, can be defined as the energy required to lift an aluminum ion from the potential well of its lattice.)

Putting this information together, one obtains the widely used expression for MTBF, as

$$\text{MTBF} = \frac{1}{\text{F.R.}} = B\ J^{-n} \exp\left(\frac{\Delta H}{kT}\right) \quad - - - (1)$$

Where $k = 8.6 \times 10^{-5}$ ev/k° (Boltzmann constant) and B and n are constants.

The value of ΔH for aluminum films used in semiconductor industry varies from 0.4 to 0.9 ev.

The parameter B depends on several physical quantities such as volume resistivity of the metal, electron mean free path, charge of the ion, a factor which represents the local material loss necessary for a failure and possibly some others. It may even have a slight dependence on temperature.

B. Thermal Fatigue Model

This model is used to calculate the number of cycles to failure for chip-to-substrate solder joints. As mentioned earlier, the failures occur because of thermal fatigue, which results from turning the machine on and off.

Plastic strain on a joint under the influence of thermal cycling is given by(6)

$$E_p = (\alpha_{sub} - \alpha_{Si})\ \frac{d}{h}\ \Delta T \quad - - - - - - - - (2)$$

where

α_{sub} = Coefficient of thermal expansion for ceramic substrate.

α_{si} = Coefficient of thermal expansion for silicon

d = Distance of the joint from the neutral or stationary point - that point on the chip which does not move relative to the substrate.

h = Height of the solder

ΔT = Temperature excursion for the cycle

For a given module every parameter except ΔT on the right side of the above equation is a constant.

$$\therefore E_p = (\text{Const.})\ \Delta T \quad - - - - - - - - - (3)$$

The Coffin-Manson equation(13) relates plastic strain and the number of cycles, N, to failure; as

$$N(E_p)^2 = (\text{Const.})$$

$$N\ (\Delta T)^2 = \text{constant} \quad - - - - - - - - - (4)$$

This result, however, needs modification to accommodate the effect of frequency and of maximum temperature of the cycle on the life of the joint. After incorporating these, one obtains(5),

$$N = C\ (\Delta T)^{-2}\ f^{1/3}\ \Phi\ (T_{max}) \quad - - - - - (5)$$

Where C is a constant, f is frequency of the cycle and Φ an empirically determined function.(14)

IV. STRESS TESTING

By stress testing we mean subjecting the devices to conditions much more severe than field use conditions; for example, powering an aluminum land or a resistor at twice or thrice the current for which it is designed and at the same time raising the temperature of the environment by 50°C to 100°C.

The chief aim of stress testing is to accelerate the modes of failure. Stress testing is the only way to determine the life of semiconductor devices because, if the tests were to be carried out under machine-use conditions, it would take millions of device test hours before one would observe a single failure.

Stress-testing accomplishes many other important objectives too. The main ones are:

(a) To project the total failure rate for the product;
(b) To uncover and study various modes of failures;
(c) To establish the quality of the aluminum film;
(d) To carry out the apportionment of the total failure rate; and
(e) To use the stress-testing as a means of qualifying a new process.

Most of the stress-testing is done on a special test pattern (or test site)(15) fabricated on the same wafer along with the actual product. A reliability test site must contain at least one test pattern for every one of the known modes of failures. But most important to the whole philosophy of test site design is that a given test pattern must accomplish only the specified function.

A. The Stress Tests and Test Conditions:

Given below are a few of the tests one needs to perform.

1. Aluminum Land:

To stress aluminum lands, three types of patterns should be used.

a. A typical stripe on the chip should be faithfully reproduced on the test site and be stressed, i.e., a line which runs over the same oxide steps at the same places and also has the same geometrical shape (turns, etc.) as the actual line.

b. Fixed length straight sections of different widths. From the result of these measurements on can drive the lives of all the small-length lines used on the chip.

c. A line running over the oxide steps, preferably open contact holes. This test would determine the degree of land thinning while passing over the oxide steps.

2. Aluminum-Silicon Contacts:

As discussed in Section II it is the positive contact of the resistor which opens up. To stress aluminum silicon contacts we should design the pattern in such a way that the maximum stress is on the contacts while under test. This can be achieved by interconnecting a large number of resistors in a series and parallel combination. The width of the interconnecting lands should be large enough as to lower their failure rate considerably below that of the contact holes for the test conditions.

3. Chip-to-Substrate Joints:

As discussed in Section II, the failure is caused by thermal fatigue due to raising and lowering of the temperature in the event the machine is turned on and off. To accelerate this failure mode in the laboratory one ought to:

(i) reduce the period of thermal cycling and
(ii) increase the range of the temperature variations.

Periods like 2 to 3 cycles per hour with a range of 0 to 100°C and 0 to 150°C have been used in the past.(6)

The joints have also been seen to fail in a mode called "delamination" where the pad as a whole comes off from the chip.

4. Corrosion Tests:

The modules (with sylgard) are subjected to an environment of high humidity and high temperature, generally 85%R.H. and 85°C. A bias of about 10 volts is put across two closely spaced lands on the chip. The test requires measurement of leakage across the lands. The magnitude of the leakage is in direct proportion to the porousity of sputtered SiO_2 above the aluminum.

B. Correlation of Stress Results to Real Life:

Once we have the results of stress-testing, the next and the most important question is to relate them to the machine-use conditions. The quantity which translates these results to machine-use conditions is called acceleration factor and is denoted by K. More precisely, K is a ratio of the failure rate for accelerated (stress) conditions to the failure rate for the real time (machine-use) conditions:

$$K = \frac{(F.R.)_A}{(F.R.)_R} = \frac{(MTBF)_R}{(MTBF)_A} \quad - - - - - (6)$$

K is a measure of the severity of the stress test.

Acceleration factor for power stress test is derived below. If J_A and T_A are the current density and temperature of the stripe under accelerated (stress) conditions and J_R and T_R are the corresponding values for real-time (machine-use) conditions, the accelerator factor is:

$$K = \frac{1}{r}\left(\frac{J_A}{J_R}\right)^n \exp.\left\{\frac{\Delta H}{k}\left(\frac{1}{T_R} - \frac{1}{T_A}\right)\right\} \quad - - - (7)$$

Where r is duty cycle for the stripe under machine-use. The only unknown in this equation is n and ΔH, which can be evaluated from the plot of stress test results on normal probability paper. Similarly one can obtain acceleration factors for thermal cycling test.

V. CONCLUSIONS

The physical models for electromigration and thermal fatique have been quite successful in predicting the results for one set of stress conditions from the results of the other set.(6,12) This is very encouraging in a situation where enough data from the field is not available to make a meaningful translation to the lab and thus confirm the stress test results.

It may be mentioned that there is no physical model for T/H test, which is widely used in the industry. Such a model will be very helpful in understanding the phenomenon and in providing improvements to the structure.

SUMMARY OF STRESS TESTS

TEST FOR	MODE OF FAILURE	PHENOMENON OR CAUSE RESPONSIBLE FOR THE FAILURE	TEST CONDITIONS
1. Aluminum Land (i) A typical line on the chip. (ii) A straight line (iii) Line over the oxide steps 2. Al-Si Contacts 3. Al Terminals	Electrical Open	Electromigration	Power Stress (High J, High T)
4. Silicon Junctions	High Leakage; Soft Break Down	High Impurities	Reverse Bias & High Temp.
5. Chip-to-Substrate Joints	Cracked Joints	Thermal Fatique	Thermal-Cycling
6. Corrosion Test	(i) Corroded Al Lands	Poor Sputtered SiO_2	High Humidity, High Temp., & Bias Across Al Lands.
	(ii) Corroded Pb-Sn Pad	Poor Sylgard Coverage	
	(iii) Leakage Across Biased Lands		
7. Chip-to-Substrate Joints	Cracked Joint		(i) Vibration (ii) Thermal Shock

Looking at the semiconductor industry as a whole, MTBF for an LSI module is in the range of three to thirty million hours. Furthermore, the failures in interconnecting aluminum metallurgy on the chip and in chip-to-substrate bonds constitute about 80 to 90% of the total. These two areas have ample room for improvement. The package in particular needs more attention.

REFERENCES

1. Electronics, April 28, 1969, page 40.

2. Attardo, M. J. and R. Rosenberg, "Electromigration Damage in Aluminum Film Conductors", presented at the 8th Annual Reliability Physics Symposium, April, 1970.

3. Blech, I. A. and Meieran, E.S., "Direct Transmission Electron Microscope Observation of Electrotransport in Aluminum Thin Films," Applied Physics Letters, 11, 263 (1967).

4. Miller, L. F., "Controlled Collapse Reflow Chip Joining," IBM J. Res. Develop. 13, 239 (1969).

5. Goldmann, L. A., "Geometric Optimization of Controlled Collapse Interconnections," IBM J. Res. Develop. 13, 266 (1969).

6. Norris, K. C. and Landzberg, A.H., "Reliability of Controlled Collapse Interconnections," IBM J. Res. Develop. 13, 266 (1969).

7. Huntington, H. B. and Grone, A. R., "Current-Induced Marker Motion in Gold Wires," J. Phys. Chem. Solids, 20, 76 (1961).

8. Grone, A. R., "Current-Induced Marker Motion in Copper," J. Phys. Chem. Solids, 20, 88 (1961).

9. Penney, R. V., "Current-Induced Mass Transport in Aluminum," J. Phys. Chem. Solids, 25, 335 (1964).

10. Huntington, H. B., and Ho, Sin-Chung, "Electromigration in Metals," J. Phys. Sco. of Japan, 18, 202 (1963).

11. Chabra, D. Ainsley, N., "Open Circuit Failures in Thin Film Conductors" IBM Components Division, East Fishkill Facility, N.Y. Tech. Report 22.419, July 1967.

12. Black, J. R., "Electromigration Failure Modes in Aluminum Metallization for Semiconductor Devices," Pro. IEEE 57, 1587 (1969).

    "Electromigration - A brief survey and some recent results," IEEE Trans. Electron Devices, ED-16, 338 (1969).

    "Mass Transport of Aluminum by Momentum Exchange with Conducting Electrons," Proc. 1967 Ann. Symp. on Reliability Physics, IEEE Cat. 7-15C58, November 1967.

    "Metallization Failures in Integrated Circuits," RADC Tech. Report TR-68-243, Oct. 1968.

13. Manson, S. S., "Thermal Stress and Low Cycle Fatique," McGraw-Hill Book Co., Inc., New York 1966, p. 132.

14. Souther Research Institute, "Fatique Properties of Solder Materials," I.D. No. 2281-59, August 10, 1960.

15. Sahni, R. J., "Use of Test Patterns in Evaluating the Reliability of Integrated Circuits", Presented at the 8th Ann. Reliability Symposium, April, 1970.

ALTERNATING LOGIC AND ITS APPLICATION TO FAULT DETECTION

H. Yamamoto, T. Watanabe and Y. Urano
Research and Development Laboratory,
Kokusai Denshin Denwa Company, Ltd.
(Japan's Overseas Radio & Cable System)
Tokyo, Japan

ABSTRACT

This paper proposes a new redundant design technique, named "alternating logic" design, which is highly effective for fault detection and diagnosis in digital computing systems. Alternating logic is a kind of duplicated system design whose redundancy is taken in the time domain in contrast with the conventional hardware duplication schemes.

Alternating logic, in other words, is based on the successive execution of a required function and its dual function. To put this alternating logic into practice, a kind of flexible logic circuit whose function produces the above two functions is required. Two practical realization methods of such a circuit are presented: one is based on self-dual function circuits and the other on majority logic elements. These methods are immediately applicable to any digital system design.

INTRODUCTION

This paper proposes a new redundancy technique for a digital computing system, which we name "alternating logic". As digital systems become larger and more complicated, it becomes very difficult to quickly detect and locate faults. Automatic fault diagnosis usually requires a large number of procedures and sophisticated and costly testing tools. In our opinion, the difficulty of this fault diagnosis depends on the property of the logic circuits or elements available. These are designed only to attain the required function at minimum cost, and without regard for potential faults.

One of the solutions for this difficulty is to devise an element-by-element "easily-diagnosable" logic circuit. Even though the new logic element contains some redundancies, it may pay for itself, if one takes into consideration the cost of fault diagnosis in the conventional digital system.

Of course, a conventional hardware duplication or triplication method is considered to some extent in accord with the above idea. But it is not sufficient. The so-called "flexible" logic circuit[1~4] seems to possess the potentiality to realize this idea.

Alternating logic is a method of duplicating a design based on the successive execution of a given function and its dual function. Thus, it can be said that the redundancy of this system is taken in the time domain in contrast with the conventional hardware duplication.

In the following sections, a formal definition of alternating logic is given. It is pointed out that the alternating logic design requires a new hardware logic circuit, whose function produces the conventional Boolean output followed in time by its dual-function output. Subsequently, it is shown that a conventional self-dual logic circuit or any kind of majority logic element may be used for this purpose, also that it is always possible to detect all possible single faults by watching the final output of a network designed with alternating logic elements. Some discussion follows concerning the practical application of this system to an "easily-diagnosable" system. Finally, we refer to an ideal case, where the alternating logic design technique is applied to a fail-safe logic system design and its realization is discussed briefly.

ALTERNATING LOGIC AND SOME DEFINITIONS

"Alternating Logic" is a logic design technique where two truth values are represented by 2-tuples whose components are different binary values, i.e., (1, 0) and (0, 1). In this paper, the two binary values forming a 2-tuple are considered as a time series; one is followed by the other, and we assume that the left component of each 2-tuple arises first. In this sense, we may say that these 2-tuples are "alternating". That is why we name this technique alternating logic. We can say that alternating logic is a kind of a repetitive double redundant system whose redundancy is taken in the time domain. The 2-tuples (0, 0) and (1, 1) are redundant in this design technique, and therefore may be used as fault indicators when a fault occurs in a circuit utilizing this technique. Hereafter we call these 2-tuples the fault signals.

It is possible to construct two kinds of alternating logic conventions. These are duals of each other, depending on how (1, 0) and (0, 1) are assigned to the truth values 1 and 0. Either 2-tuples could be assigned to the value 1, but we shall adopt the convention that $1 \sim (1, 0)$ and $0 \sim (0, 1)$.

Thus, for a given logic function $f(x_1, \ldots, x_n)$, an alternating logic design associated with f is defined as follows:

Definition 1.

An alternating logic design associated with $f(x_1, \dots, x_n)$ is a design in which each input variable associated with x_i is represented by a 2-tuple $(x_i, \bar{x}_i)$ and the output variable by $(f, \bar{f})$ respectively. $(i = 1, \dots, n)$

Hereafter we shall call these 2-tuples alternating variables or signals. Alternating variables that correspond to inputs are called alternating inputs and those that correspond to an output is called an alternating output.

Definition 2.

An alternating function associated with f is a function which yields an alternating output $(f, \bar{f})$ for a certain set of alternating inputs

$$\{(x_i, \bar{x}_i)\}, \quad (i = 1, \dots, n).$$

It is denoted by F_f.

Definition 3.

An alternating circuit is a hardware circuit which realizes an alternating function (we use the same symbol F_f to show the alternating circuit as the alternating function).

Definition 4.

Fundamental alternating circuits or alternating elements are fundamental logic circuits or elements, which realize alternating functions and from which an alternating circuit may be composed for any arbitrary logic function.

FLEXIBLE LOGIC CIRCUIT MODEL OF ALTERNATING LOGIC

From the definitions in the preceding section it can be shown that if a logic circuit exists whose function is interchangeable with a given function f, and its dual function f^D, the conditions for attaining an alternating circuit F_f are met. This interchanging must be performed synchronously with the alternating of two components of alternating inputs to attain an alternating logic design.

If this condition is satisfied, the second component of the output 2-tuple for the alternating inputs imposed on the flexible logic circuit is as follows:

$$f^D(\bar{x}_1, \dots, \bar{x}_n) = \bar{f}(x_1, \dots, x_n).$$

Thus, the circuit yields an alternating output $(f, \bar{f})$ for the alternating inputs imposed. In this sense, we write

$$F_f \equiv (f, f^D)$$

or

$$F_f(X_1, \dots, X_n) = (f(x_1, \dots, x_n), f^D(x'_1, \dots, x'_n))$$

where $x_i \equiv (x_i, x'_i)$, $(i = 1, \dots, n)$ and $x'_i = \bar{x}_i$ in the case where no failure exists in the system.

An alternating function F_f is shown schematically in Figure 1. Figure 2 shows a model of the flexible logic circuit realizing the alternating function F_f. In Figure 2 the additional terminal C is to be fed a controlling signal c, which controls the alternating of its functions.

More specifically, the function f_c becomes f or f^D in the following manner;

if $c = 1$, $\quad f_c \equiv f_1 = f$,

and

if $c = 0$, $\quad f_c \equiv f_0 = f^D$.

Figure 1. Alternating function F_f associated with $f(x_1, \dots, x_n)$

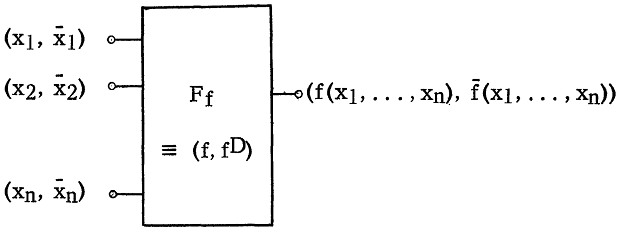
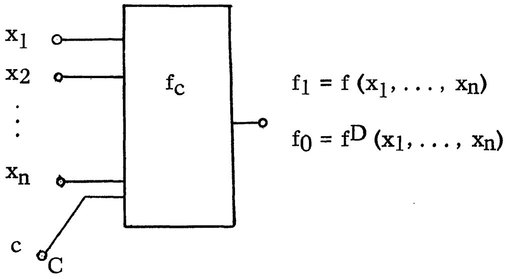


Figure 2. Hardware model of alternating circuit



For this, we write F_f as

$$F_f = (f, f^D) = (f_1, f_0) \equiv [f_c]_{(1,0)}.$$

This expression shows that if the circuit f_c is available, the alternating function F_f is realized by feeding an alternating constant (1, 0), where 1 is followed by 0, to the terminal C. Using the same circuit f_c, the dual function of F_f, denoted by $(F_f)^D$ is realized as follows:

$$(F_f)^D = (f^D, (f^D)^D) = (f^D, f) = (f_0, f_1)$$
$$= [f_c]\ (0, 1) .$$

It is natural to assume the existence of the controlling signal which controls the alternating of the functions. Hence, it is reasonable to adopt the circuit f_c as a model of the hardware realization of an alternating function F_f. This model can well be applied not only to a general circuit but also to a fundamental alternating circuit.

ON THE PRACTICAL REALIZATION OF ALTERNATING FUNCTIONS

The authors propose the alternating logic design technique as a new redundancy design technique that is effective in detecting and locating faults in a large and complicated digital system. To make it practically available, the circuit f_c must be fabricated. As far as the function is concerned, it is possible to fabricate using any available logic gate. In fact, a circuit realizing

$$g(c, x_1, \ldots, x_n) = c \cdot f(x_1, \ldots, x_n) + \bar{c} \cdot \bar{f}(\bar{x}_1, \ldots, \bar{x}_n)$$

can be adopted as f_c, provided that the variable c is taken to be the controlling signal. An important point to be taken into consideration for its practical use is the redundancy required to realize the circuit f_c.

Specifically speaking, if the circuit f_c can be fabricated by using as many logic elements as those of a circuit realizing the function f, then it can be adopted for practical use without any hesitation. However, if the circuit f_c needs many more elements than f, the alternating logic design of the circuit f_c becomes less efficient and less desirable to use. In this case, a well-known hardware duplication method[5, 6] is probably more efficient than the alternating logic method, because the former does not "waste" any time while the 2-tuple latter does.

Though it remains to be solved how to devise a new and more efficient circuit for f_c, we can point out two practical examples of hardware realization of an alternating logic design which are available at present. One is an arbitrary self-dual function circuit composed of any kind of conventional logic elements. The other is an arbitrary logic circuit composed of any kind of majority logic elements such as a tunnel diode pair or a parametron, etc.

In the following sections we shall discuss the effectiveness of these circuits and elements as alternating circuits and fault detecting means.

SELF-DUAL FUNCTION CIRCUIT

Theorem 1.

If a function $f(X)$* is self-dual, then $f(X)$ itself is an alternating function F_f, associated with $f(X)$.

Proof. By the self-duality of $f(X)$, $f(\bar{X}) = \bar{f}(X)$. Thus for alternating inputs $(X, \bar{X})$**, it is evident that an alternating output $(f, \bar{f})$ is given by $f(X)$.

Q. E. D.

As shown in the above theorem, any circuit whose function is self-dual can serve itself as its own alternating function without employing any additional devices provided that it is being fed with alternating inputs.

Although self-dual functions are just a portion of all possible functions, an arbitrary function can be represented as a self-dual function with at most one redundant variable added to the input variables in the given function.

Theorem 2.

An arbitrary function of n variables is represented as a self-dual function of n' variables by adding at most one redundant variable, where $n' \geq n + 1$.

Proof. For an arbitrary function $f(X)$ which is not self-dual, we make a function $f_*(X, x_{n+1})$ as follows:

$$f_*(X, a) = f(X),$$
$$f_*(X, \bar{a}) = \bar{f}(\bar{X}),$$
where $a = 1$ or 0.

Then, $\bar{f}_*(\bar{X}, \bar{a}) = f(X) = f_*(X, a)$.

Thus, $f_*(X, x_{n+1})$ is self-dual.

Q. E. D.

The above theorem shows that it is possible to fabricate an alternating circuit practically for an arbitrary given function. Using x_{n+1} as c in the function f_c, the function f_* becomes just the same as f_c. In other words, the circuit f_* can be the variable logic circuit whose function is interchangeable with f and f^D, by feeding a control signal to the terminal for the additional variable x_{n+1}.

Now we have to know what effect to expect when a fault is caused in the circuit realizing a self-dual function. As far as the function is concerned, any self-dual circuit serves as an alternating function, regardless of the logic elements composing the circuit. However, to realize the capability of detecting faults in this alternating scheme, some restrictions on the structure of the circuit are required. Fortunately, in spite of these restrictions, it is always possible to fabricate a circuit for an arbitrary function, such that all possible single faults can be detected by watching its output signal. The following theorem states it formally.

---

* $X \equiv (x_1, \ldots, x_n)$, $\bar{X} \equiv (\bar{x}_1, \ldots, \bar{x}_n)$

** $(X, \bar{X}) \equiv \{(x_1, \bar{x}_1), \ldots, (x_n, \bar{x}_n)\}$

Theorem 3.

Let us adopt a logic circuit to realize a given self-dual function that is a 3-staged circuit consisting of conventional AND, OR and NOT gates, where the NOT gates are located only at the first stage (input side). When an arbitrary gate in the circuit fails, the output for any alternating input is either a logically correct[*] alternating output, or a fault signal, i. e. (0, 0) or (1, 1). We assume that the fault is the state 1 or 0, at which the output of a gate is stuck, regardless of input states.

Proof. Let the output function of the faulty gate be $g_f(X)$; $f(X)$ is then:

$$f(X) = f'(X, g_f(X)) .$$

Evidently from the network structure, f' is "positive"[**] concerning g_f , so

$$f'(X, 0) \overset{***}{\leq} f'(X, g_f(X) \leq f'(X, 1).$$

Let us write the output function of the failed circuit as f^1 (f^0) , when the output of the gate g_f is stuck at 1 (0), and then

$$f^1 = f'(X, 1),$$

and

$$f^0 = f'(X, 0).$$

Hence we have the following inequalities for the output of the circuit.

$$(f^0(X), f^0(\bar{X})) \leq (f(X), f(\bar{X})) \leq (f^1(X), f^1(\bar{X})).$$

Thus we can conclude that the failed output

$$(f^0(X), f^0(\bar{X})) \; [(f^1(X), f^1(\bar{X}))]$$

becomes either (0, 0) [(1, 1)], indicating a faulty state of the circuit, or it is equal to

$$(f(X), f(\bar{X})) = (f(X), \bar{f}(X))$$

i. e. a logically correct[*] alternating output.

Q. E. D.

For multiple faults, similar conclusions on their detection may be proven; but we shall not cover this in this paper. Suffice it to say here that this method is practical and effective for most multiple faults in many conventional self-dual circuits.

SYNTHESIS OF AN ALTERNATING CIRCUIT FROM AN ALTERNATING ELEMENT AND ITS FAULT DETECTABILITY

In this section, we intend to show that an arbitrary logic circuit composed of majority logic elements, such as tunnel diode pairs or parametrons, etc., does work as the alternating circuit associated with the function of this given circuit, and that each majority logic element can be considered as an alternating element. Subsequently, the detectability of faults occurring in the circuit implementation will be discussed.

A majority function is a typical self-dual function, and it becomes an alternating function F_{OR} (F_{AND}) associated with OR (AND) , if an alternating constant (1, 0) ((0, 1)) is fed to one of the input terminals synchronously with the other alternating inputs. Logic negation NOT is also a self-dual function, so it can be used as an alternating function F_N associated with NOT. These three alternating functions are sufficient to design a complete Boolean system. Thus, the conventional majority logic elements and the negation element may also be used as fundamental circuits in our alternating logic design system.

Notation

For convenience, we adopt the following notations:

$$\left.\begin{array}{l} F_{OR} \equiv (+, \cdot), \quad F_{AND} \equiv (\cdot , +) \\ F_N \equiv (- , -) \end{array}\right\} \qquad (1)$$

where + , · , and - are conventional logic sum, product, and negation respectively. And for input 2-tuples $X \equiv (x, x')$ and $Y \equiv (y, y')$, we write

$$\left.\begin{array}{l} F_{OR}(X, Y) = (x+y, x' \cdot y'), \\ F_{AND}(X, Y) = (x \cdot y, x'+y'), \\ F_N(X) = (\bar{x}, \bar{x}') . \end{array}\right\} \qquad (2)$$

Synthesis of an Alternating Circuit

The following theorem concerning the duality of a composite function is well-known and evident from the published results.[7,8]

---

[*] In this paper, for certain inputs the output of a failed circuit is said to be logically correct if and only if the output is the same as that of its normal function.

[**] Monotonically increasing.

[***] $\leq$ is defined as follows: If and only if $x_1 \leq y_1$ and $x_2 \leq y_2$, then

$$(x_1 , x_2) \leq (y_1 , y_2).$$

Theorem 4.

For any composite function which is composed of elementary functions f_{s1} , f_{s2} , ..., f_{sm} , its dual function f^D is obtained by substituting each f_{si} by the corresponding dual function f^D_{si}

$(i = 1, \ldots, m)$.

Proof. cf. appendix.

Evidently from the above theorem, a circuit consisting of the fundamental alternating elements mentioned above is also an alternating circuit. Thus, to synthesize an alternating circuit for a given function, it is sufficient to design a conventional circuit for the given function with the usual AND, OR and NOT gates and afterwards to substitute every gate with the corresponding alternating element.

Fault Mode

We assume that the possible faults which may occur in an alternating circuit are confined to the following two states of the alternating elements:

state 1. The output of the element gets stuck at 1 or 0.

state 2. The function of the element gets stuck at OR or AND (this state can occur only to F_{OR} and F_{AND}).

According to the assumption, a failed element of state 1 gives the fault signal (0, 0) or (1, 1) for any alternating input. The function of failed elements of state 2 denoted by F_{ea} and F_{eo} can be written similarly to (1) and (2) as :

$$F_{ea} \equiv (\cdot\ ,\ \cdot)\ ,\quad F_{eo} \equiv (+\ ,\ +) \tag{3}$$

or

$$\left.\begin{aligned} F_{ea}(X, Y) &= (x\cdot y,\ x'\cdot y')\ , \\ F_{eo}(X, Y) &= (x+y,\ x'+y'). \end{aligned}\right\} \tag{4}$$

Evidently from the above expressions, the output of F_{ea} (F_{eo}) is (0, 0) ((1, 1)), when two alternating inputs X and Y are different from each other. If the two inputs are the same, the output gives the same alternating signal as the alternating input, which is logically correct.

As a result, if one of the fundamental alternating elements fails, it outputs (0, 0) or (1, 1) for some alternating input fed to the circuit, and this fault signal is fed to the subsequent normal elements. To discuss the fault detection by means of this fault signal, we now have to know how the fault signal is transferred to the final output terminal of the circuit. We know the expression (2) is valid for arbitrary input 2-tuples as well as alternating inputs. Therefore, using this expression, we know the responses of the alternating elements to all possible inputs including the fault signals. The results are shown below, where, for simplicity, the symbols +, · , - are used instead of F_{OR}, F_{AND}, FN respectively, and in addition we write

$$(1,0) \equiv 1,\ (0,1) \equiv 0\quad (0,0) \equiv \phi_0\quad (1,1) \equiv \phi_1\ .$$

$$\left.\begin{aligned} &1+0=0+1=1+1=1,0+0=0 \\ &1\cdot1=1,\ \ 0\cdot1=1\cdot0=0\cdot0=0 \end{aligned}\right\} \tag{5}$$

$$\phi_i+0=0+\phi_i=1\cdot\phi_i=\phi_i\cdot1=\phi_i \tag{6}$$

$$\phi_i+1=1+\phi_i=1,\ \ 0\cdot\phi_i=\phi_i\cdot0=0 \tag{7}$$

$$\left.\begin{aligned} &\phi_i+\phi_i=\phi_i\cdot\phi_i=\phi_i \quad \cdots\cdots\cdots\cdots\cdots\cdots a \\ &\phi_i+\phi_j=1,\ \ \phi_i\cdot\phi_j=0(i\neq j)\cdots\cdots\cdots\cdots b \end{aligned}\right\} \tag{8}$$

where i, j = 1 or 0.

Statement (5) shows that the alternating logic rules correspond to the conventional Boolean rules in a failure-free design. Statements (6) and (7) correspond to the case where only one input fails. In this case, the output is the fault signal or the logically correct alternating output according to the other input value. Statement (8) shows the case where both of the inputs are faulty. In this case, if both inputs are given with the same fault signal, the fault signal is transferred to the final output, but otherwise an alternating output which may be logically incorrect is generated.

Fault Detection

The circuit treated here can be any circuit, as long as it is synthesized from alternating elements.

We assume that fault detection is performed only by detecting the fault signal at the final output terminal of the circuit.

For a fault occurring to one of the elements in the circuit, we say that the fault is detectable if a fault signal can be detected for at least one input for the circuit, and otherwise it is undetectable.

It is possible to construct a circuit with multiple faults which are undetectable. For example if two elements directly preceding an element fail at the same time, but differently (i. e. one generates (0, 0) and the other (1, 1)), then no fault signal will be outputted as shown in (8)-b. This is undesirable, but, in most cases, the detection of a single fault is most important and has practical usage, if it can be performed sufficiently fast. In the following two theorems, we shall show that alternating logic design is very effective for single fault detection.

Theorem 5.

In an arbitrary alternating circuit consisting of alternating elements, if one of the alternating elements fails, then the output of the circuit is either a fault signal ((1, 1) or (0, 0)) or a logically correct alternating output.

Proof. For all possible alternating inputs, the output of any alternating circuit is always an alternating output. Any partial circuit of the circuit in question is evidently an alternating circuit according to Theorem 4. Hence, it follows that the output of every alternating element in the circuit is an alternating output without exception, when there is no fault in the circuit. If a certain alternating element in the circuit fails and produces a fault signal ((0, 0) or (1, 1)) for some alternating inputs, either of the two components of the fault signal (2-tuple) is logically correct. In other words, it produces either of the two components of the alternating output of the element in the case where there is no fault in the circuit. Therefore, that component of the final output (2-tuple) of the circuit which corresponds to the logically correct component of the output of the failed element is necessarily logically correct.
Namely, even though the final alternating output of the circuit may become faulty, this fault can occur only to one component of the output (2-tuple). Thus, the final output of the circuit is either a fault signal or a logically correct alternating output. Q. E. D.

Theorem 5 is not trivial, because, if some loop connections exist among the alternating elements, it is possible to conjecture the case where a fault signal of a single fault is transferred through two different routes composing the loop and is fed to both inputs of another element terminating the loop as two different fault signals, viz., (0, 0) and (1, 1). Thus, as shown in (8)-b, the element terminating the loop might prevent the fault signals from flowing to the final output of the circuit and thereby generate a logically incorrect alternating output. By this theorem it is proven that such a case cannot exist.

Definition 5.

An alternating element in an alternating circuit is redundant, if and only if the output of the circuit is not affected at all by forcing the output of the alternating element to either 0 or 1.

Definition 6.

An alternating circuit is redundant, if and only if it contains at least one redundant alternating element, otherwise it is nonredundant.

Theorem 6.

A single fault in any alternating element which is part of a nonredundant alternating circuit is always detectable.

Proof. Assume that there exists an undetectable single fault in a given nonredundant alternating circuit. Then all the possible outputs of the circuit are alternating and logically correct according to theorem 5. Thus the faulty element does not affect the normal function of the circuit at all. Namely the faulty element is redundant. This is a contradiction. Q. E. D.

By the above two theorems, we can convince ourselves that as far as an arbitrary nonredundant circuit is concerned, any single fault is successfully detectable, and that whenever an output is alternating, it is logically correct.

It should be realized, however, that a fault signal is not necessarily transferred to the final output for all possible inputs. On this point, Theorem 6 states that there exists at least one input, for which the fault signal is transferred to the final output. Hence, in order to perform fault detection efficiently, it is important to find the minimum number of test inputs which can transfer the fault signals to the final output for all the faults expected. The number of such test inputs is much smaller, in most practical examples, than in conventional logic circuits. This is mainly because in this alternating design scheme, a test input implies two conventional design test inputs which are duals of each other.

In order to avoid multiple faults, it is necessary to detect a fault before another one arises. In practice, it is sufficient to feed such test inputs to the circuit in much shorter intervals than the meantime between failures (MTBF) of the design. Or, if the expected inputs during the use of the design are considered as including a sufficient number of the test inputs statistically, then it is unnecessary to resort to any special testing tactics. We have only to incorporate a fault signal detecting device in the circuit to monitor the final output.

In the preceding discussions, three functions F_{OR}, F_{AND} and F_N are adopted as the fundamental alternating functions. The F_{OR} and F_{AND} can be realized by any majority logic element with alternating constants (1, 0) and (0, 1) fed to one of its inputs respectively. Although the above three functions are sufficient to make a complete Boolean system, another function F_M (associated with majority logic functions) should be added to those three fundamental functions in order to design a given logical circuit with a smaller number of elements. This is possible, because F_M is realized by any majority logic element only with alternating inputs fed to its inputs. As follows from the preceding discussions, the theorems 4, 5, and 6 can be applied to any alternating logic system which is based on these four fundamental alternating functions. It also follows that any conventional logic circuit only composed of majority logic elements may be used for an alternating logic design, if only its conventional inputs and constant inputs are substituted for the corresponding alternating inputs and constants respectively. In this case, no additional hardware, with the exception of the fault signal detector, is needed. So, we can conclude that the redundancy of this scheme is taken truly in the time domain, and that this redundancy realizes fault detectability.

Internal Diagnosis of Faults

The output of any partial circuit of an alternating logic circuit is always alternating when there is no fault in the circuit. Therefore, it, too, may be used as a diagnosing output, if necessary, just like the final output. Thus, internal diagnosis may be performed very simply.

Improving the Time Redundancy

Since the alternating logic design is based on the successive execution of a function and its dual function, it takes at least twice the time needed to perform the same function in the conventional manner. It is, however, possible to make the time redundancy smaller by operating the system usually in the conventional manner and using the alternating logic scheme only during the diagnosing phase. This takes away only a portion of the working time of the design.

FAIL-SAFE ALTERNATING LOGIC

In this section we shall discuss an ideal circuit in relation to a fail-safe logic design.

In a conventional Boolean design, a fail-safe logic design is defined as a design in which the final output is confined to either of the two truth values for any fault occurring in the circuit.

A similar definition is adopted here for a fail-safe alternating logic design.

Definition 7.

> An alternating logic design is fail-safe, if and only if the output of the system is either a logically correct alternating output or a fault signal (0, 0) or (1, 1) for any fault occurring in the design.

We can say that the alternating logic discussed in preceding sections is fail-safe only for a single fault at a time. It is not fail-safe for the general multiple faults case. This is a natural result of the fact that the design is a duplicated design by itself.

It is possible to prove, however, that a fail-safe alternating logic design which is fail-safe for all possible multiple faults can be realized, provided that an ideal alternating element can be constructed which outputs either of the two truth values as a result of a fault. The detailed theoretical results obtained concerning the synthesis of such a fail-safe alternating logic element is beyond the scope of this paper. Although it is difficult to devise an ideal alternating element, a fail-safe alternating logic design based on such an element would be very valuable for use in an "easily-diagnosable" system design. The authors would like to state that a fail-safe alternating element is not just a dream. A signal controlling device utilizing similar fail-safe parametrons is being tested by JNR* Research Laboratory.[9]

CONCLUSIONS

In this paper, "alternating logic" a new redundancy design technique for digital computing systems, is proposed, and its effectiveness in fault detection is shown. Although our discussions are focused on those faults which occur to hardware logic elements in a design, one can easily conclude that alternating logic is also effective in detecting intermittent errors which might occur to internal signals in the circuit, or even to input signals.

Alternating logic may be said to be a kind of duplication technique, in which duplication is based on the successive execution of a required function and its dual function. A simple repetition of a required function cannot realize the same fault detecting effect. The combination of a function and its dual function alone provide this property.

The redundancy of this design is taken in the time domain in contrast with the conventional hardware duplication design technique. Yet fault detection in this redundancy presents the same effect as that of hardware duplication.

The principal merits of the alternating logic are:

1. The hardware cost of this system is expected to be cheaper than that of the conventional hardware duplication technique (ideally much the same as the corresponding nonredundant design).

2. The fault signals are coded differently from the correct binary signals, thus fault detection may be performed very simply only by watching the final output of the design without any complicated testing tactics and tools which would be needed for a conventional digital design.

3. Fault locating, or diagnosing internally to the design, may be performed easily by making available necessary internal testing terminals.

From 2 and 3 above, one can conclude that this design technique applied results in "easily-diagnosable" system designs.

---

* Japanese National Railways

Two kinds of hardware realization methods for alternating logic are presented in this paper: one is based on a self-dual function circuit and the other on majority logic elements. Both methods can be immediately applied to any practical digital design. However, to make alternating logic widely applicable, more efficient and appropriate elements should be devised. Of course, the authors are trying to find such elements.

ACKNOWLEDGEMENTS

The authors wish to thank Dr. H. Shinkawa, Dr. S. Ohshima, Dr. Y. Nakagome, and Dr. S. Inoue for their encouragement and permission to present the paper. Mr. Y. Koseki and Mr. T. Ichikawa read and criticized the manuscript.

Additional Remarks*

Bark and Kinne proposed fundamentally the same idea as the "alternating logic" early in 1953.[10] However, they did not enter into such fault detectability as clarified in this paper.

REFERENCES

1. Fukui, K. and Naemura, K., "Standardization of Integrated Circuits," J. IECEJ**49, 628-632 (1966).
2. Kiyasu, Z. and Maruoka, A., "On Flexible Logical Cells," IECEJ Trans. 52-C, 319-326 (1969).
3. Maitra, K. K., "Cascaded Switching Networks of Two-Input Flexible Cells," IRE Trans. EC-11, 136-143 (1963).
4. Minic, R. C., "Cutpoint Cellular Logic," IEEE Trans. EC-13, 685-698 (1964).
5. James, D. C., Kent, A. H. and Holloway, A., "Redundancy and the Detection of First Failure," IRE Trans. RQC-11, 8-29 (Oct. 1962).
6. Sasaki, M., "Redundancy," J. IECEJ 47, 1609-1617 (1964).
7. Takahashi, H., Parametron Keisanki (Parametron Computer) (Iwanami Book Co., Inc., Tokyo, Japan 1968), 1st ed. Chap. 5, p. 259.
8. Kiyasu, Z., Digital Kairo no Sugaku (Mathematics for Digital Circuits) (Kyoritsu Book Co., Inc., Tokyo, Japan 1960), 1st ed. Chap. 5, p. 47.
9. Okumura, I. et al., "A Signal Controlling Device for Railways using Parametron," Report of Technical Group on Electronic Computer of IECEJ, no. EC-67-5 (1967).
10. Bark, A. and Kinne, C. B., "The Application of Pulse Position Modulation to Digital Computers," Proc. National Electronics Conference, 656-664 (Sept. 1953).

APPENDIX: PROOF OF THEOREM 4

An example of the proof is presented here.

To begin with, the structure of a composite function must be defined. We consider that a given composite function is multi-staged. For the convenience let us renumber all the elementary functions, and denote the i-th elementary function at the k-th stage as f_{ski}.

($1 \leq i \leq N_k$, where N_k is the number of elementary functions at the k-th stage.)

We call input variables to a composite function initial inputs, denoted as x_i ($i = 1, 2, \ldots, M$).

Thus, a composite function is defined as a n-staged composite function whose elementary function at each stage are wired as follows:

1. Inputs of any elementary function at the first stage are initial inputs alone.
   Namely,
   $$f_{s1i} \equiv f_{s1i}(x_1, \ldots, x_M).$$
2. Inputs of any elementary function at the k-th stage are both initial inputs and outputs of elementary functions at the k'-th stage, where $k' \leq k-1$ and $2 \leq k \leq n$.
   Namely,
   $$f_{ski} \equiv f_{ski}(x_1, \ldots, x_M, f_{s11}, \ldots, f_{s(k-1)N_{k-1}}).$$
3. The output of the composite function, denoted as $f(x_1, \ldots, x_M)$, is the output of an elementary function at the n-th stage.
   (We consider the number of the output of a given composite function is one, so only one elementary function exists at the n-th stage.)

The above definition implies all necessary wirings that give any reasonable composite function. (Exactly speaking, we assume implicitly that the composition is combinational.)

The output of each elementary function f_{ski} becomes a function of initial inputs alone, denoted as

---

* The authors knew Bark and Kinne's work[10] by the communication of Mr. R. J. Preiss, Diagnosable Design and Diagnosis Session Chairman of this conference, when the final manuscript of this paper was almost completed.

** IECEJ is the abbreviation of The Institute of Electronics and Communication Engineers of Japan. English versions of the journal and transaction of IECEJ are published by IEEE in U. S. A.

$g_{ki}(x_1, \ldots, x_M)$, in a given composite function.
Thus,

$$\begin{aligned} f(x_1, \ldots, x_M) &= g_{n1}(x_1, \ldots, x_M) \\ &= f_{sn1}(x_1, \ldots, x_M, g_{11}, \ldots, \\ &\qquad g_{(n-1)N_{n-1}}). \qquad \text{(al)} \end{aligned}$$

Now the proof of theorem 4 can be presented. A mathematical induction on n is applicable to this case.

Proof

Basis: The theorem is true for $n=1$.

Induction: Assume that the theorem is true for $n \leq k$. For $n=k+1$, the output of the composite function, denoted as f', whose elementary functions are all substituted by their corresponding dual functions can be written as:

$f' = f^D_{sn1}(\ldots x_i \ldots f^D_{sk'j} \ldots .)$, where $k' \leq k$.

Evidently, the output of every $f^D_{sk'j}$ $(k' \leq k)$ is also a composite function, for which the theorem holds according to the assumption.
Thus,

$$f^D_{sk'j} = g^D_{k'j}(\ldots x_i \ldots).$$

Then,

$$f' = f^D_{snl}(\ldots x_i \ldots g^D_{k'j}(\ldots x_i \ldots) \ldots).$$

By the definition of a dual function, the above statement becomes

$$\begin{aligned} f' &= \bar{f}_{snl}(\ldots \bar{x}_i \ldots g_{k'j}(\ldots \bar{x}_i \ldots) \ldots) \\ &= \bar{g}_{n1}(\ldots \bar{x}_i \ldots). \end{aligned}$$

Comparing the above statement with (al), finally we obtain

$$f' = g^D_{n1}(\ldots x_i \ldots) = f^D(\ldots x_i \ldots).$$

Namely, the theorem is true for $n=k+1$.

Q. E. D.

ERROR-FREE DECODING FOR FAILURE-TOLERANT MEMORIES*

W. C. Carter, D. C. Jessep and A. Wadia

International Business Machines Corporation
Thomas J. Watson Research Center
Yorktown Heights, New York 10598

ABSTRACT

The relative unreliability of present day memories are causing architects to use error correction/detection (EC/D) coding for memory words. The increase in the size of memory words together with the EC/D has increased the EC/D circuit failure probability so that it is comparable to the probability of memory double failures. Thus, if single error correcting and double error detecting codes are used, single circuit failure detection capability is necessary. This paper deals with the selection of memory organization and codes, and the design of the associated circuitry so that all single bit memory word errors are corrected, all double bit memory word errors are detected, all single component failures in the error correction and decoding circuitry are detected, and undetected erroneous operation may only take place if three or more failures occur in the memory array, or (exclusively) two occur in memory and one or more occur in the circuit, or (exclusively) two or more failures occur in the circuit.

1. INTRODUCTION

Samuel Johnson said "The true art of memory is the art of attention." This statement has special significance when examined in the context of error-free decoding for failure-tolerant memories. The goal of the designer, in this case, is to retrieve correct information from a memory and convert it, without error, into a form readily usable by the machine for other purposes. To achieve this goal, the designer must be attentive to the organizational structure of the memory, the coding used with the information, the circuitry used for the encoding and decoding processes, and the desired resultant form for the information as it is to be used elsewhere in the machine. To minimize the erroneous effects of a failure in a memory, the designer will choose an organization in which a single failure will affect, at most, one bit of the retrieved word and two failures will affect, at most, two bits of the retrieved word. Thus, the reliability will be enhanced by using the class of codes known as single-error-correcting, double-error-detecting (SEC/DED) codes. This paper discusses a self-testable implementation of SEC/DED decoders and encoders, or translators. Self-checking units are those which will detect the occurrence of any single component failure without previously producing an erroneous result and self-testing units are those which will test the proper functioning of every component during normal operation.

The translator, in self-testable implementation, will correct single bit errors and detect double bit errors in the memory word and will withstand a circuit failure in the translator without transmitting erroneous data from the memory.

2. MEMORY ORGANIZATION

The historical precedence for encoding memory words dates back to the Modulo 31 coding used in the RAYDAC machine, through parity encoding to the SEC/DED encodings used today. In this paper, the memory translation circuitry will not only imply that used for SEC/DED decoding (including correction) and encoding, but all additional circuitry required to transform the data bits to parity encoding for buss transmission.

2.1 Main Store Structure

By examining the basic sources of error, within an ordinary 3D memory, it can be seen that, while failure of a single memory cell or sense circuit means an error in a single position in a word, all bits in one or more words may be in error due to word-drive circuitry failures. To offset potential errors due to these diverse types of failure, Allen [1] proposed that a physically distinct memory be used for each bit of the word. That is, for a memory word of n-total bits, there will be n distinct submemories. This memory organization is referred to as the "Bit per Basic Operating Memory" (or Bit/BOM) organization. Every bit of the word is retrieved independently since each BOM used is totally isolated electrically from all other BOM's and contains its own address decoders, drivers, and sensing circuitry. The obvious advantage of such an organization is that single failures in either the cell, the sense circuitry, or the drive circuitry can now be constrained to affect a single bit of the data word at most. Two errors in a memory word require that error-inducing failures exist in two BOM's with independent groups of components.

---

*This work was supported in part by NASA under NASA Contract NAS 8-24883.

It is possible to implement a non-self-testable SEC/DED coding scheme for a 32-bit wide memory word with an equivalent gate count of 1050 circuits. Here "equivalent" means that a single exclusive-OR (XOR) operation requires four gates for its implementation as will a single register bit position. A 64-bit wide memory word with the above coding scheme will require 2300 equivalent gates and a 128-bit wide memory word will require 4950 equivalent gates. The probability that this circuitry fails is thus approximately equal to the probability that double errors occur in a 1 to 2 million byte memory array. Neither probability is negligible, especially for high reliability computer systems. Since the probability of single failures in such a memory array is higher, a SEC/DED error correction circuit will be used with a Bit/BOM memory organization. For a balanced design, the circuit failures must also be detected, and hence, the translation circuitry will be made self-testable as far as possible.

2.2 Translator Failure Analysis

The theory of designing self-checking, and self-testing circuitry for computers has been discussed previously [2, 3, 4]. This theory rests upon encoding all data and control signals and designing so that no signal line has a constant value during normal operation. The precise definition of this concept involves specifying the code space as the set of circuit signals for error-free computer operation, and the error space as the complement of the code space [2].

The data word, as categorized by the SEC/DED circuitry will have one of three possible statuses: it will contain no errors, a single error, or a double error. The new problem to be faced by the error-detection circuitry is to differentiate between single errors in the input data and single circuit failures in the translator which cause errors. A first approach is to consider the space of no input errors as code space and attempt to design SEC/DED translators which are self-checking and self-testable.

Specific circuits, which have been proved to be self-testing, will be used in the translator SEC/DED design. Figure 1 shows a two-output parity check circuit or syndrome generator circuit. With odd parity, the outputs are $\begin{Bmatrix}0 & 1\\ 1 & 0\end{Bmatrix}$ in code space and $\begin{Bmatrix}0 & 0\\ 1 & 1\end{Bmatrix}$ otherwise. Since each circuit has all combinations of variables applied to its inputs, each is extensively tested and the circuit is self-testable.

The second self-tested circuit is a Reduction Circuit for Checker Outputs (RCCO) [2] Figure 2. This circuit will be used in the translator to reduce r pairs of syndrome lines to a single pair of lines with values $\begin{Bmatrix}0 & 1\\ 1 & 0\end{Bmatrix}$, if all r pairs are in code space, and $\begin{Bmatrix}0 & 0\\ 1 & 1\end{Bmatrix}$, if one or more of the r pairs are in error space. An RCCO circuit is shown for $r = 2$ in Figure 2. Such circuits can be interconnected and cascaded in tree form to reduce an arbitrary number of self-testing line pairs to a single pair of self-testing lines for ease of monitoring the original set.

For a system consisting of interconnected modules to be self-testing, it is necessary and sufficient that each module be testable in the space defined by its inputs during the normal operation of the system. The first step is, hence, to design each module to be self-testing by techniques given in [2, 4]. However, when modules are designed to be self-testing as isolated units, their interconnection, to form a part of a larger system, does not have to be self-testable. As the inputs of one module are outputs of another module, they are no longer independent (unless specifically designed so) and all input patterns required for the testing of the successor module may not appear during normal operation of the system. It is necessary, in order to ensure testability, to make the outputs of modules that feed into another as independent as possible.

In conventional SEC/DED circuitry, many lines in the correction and detection circuitry do not change logic states over this code space, and so have constant values. Using conventional self-testing circuits, doubling the number of these lines, and making sure that the line pairs changed states over the code space resulted in too costly logic functions.

Further analysis showed that the ultra-reliable SEC/DED translator circuitry could be divided into two parts, the double input error and circuit error-detecting circuitry and the single input error-correcting circuitry. The error-detecting circuitry must be operable at all times, and should be self-testing since the probability of circuit failure and double input data error are of the same magnitude. The error-correcting circuitry must be completely self-checking over both code space and single input error spaces.

With the restriction of implementing the translator with a reasonable amount of circuitry, not all of it can be self-testing over code space. So the further requirement was added that the failure of one or more components that are untested over code space must not change the completely self-checking feature over code and single error space and that any component failure must be ultimately detectable, either by itself or in a combination of failures, over code or single error space.

This was achieved by using the self-testing circuitry discussed previously, by designing new self-checking circuitry, and by choosing the SEC/DED code so that all circuit failures which are not detected over code space are either not harmful or finally result in a detectable error when the input from the memory contains a single error.

3. TRANSLATOR DESIGN

In this section the design of the self-testing error detection circuits and the self-checking correction circuits will be described. The self-testing and self-checking properties will be stated, and mathematical proofs are given in the Appendix.

3.1 Properties of the SEC/DED Code

The notation of standard coding theory will be used in this description: n bits comprise the memory word, with k data (or message) bits d_j, and r check bits c_i. The original ideas of Hamming [6] for an SEC/DED code will be modified for this paper. In the usual parity check matrix, the implementation of the row of all 1's, which corresponds to double-error detection, requires approximately twice as many inputs (or XOR gates) as the implementing of any other row. R. A. Henle [7] suggested that this number could be halved by specifying each column with an odd number of 1's. The double-error detection property would be retained since a double input error would result in an even number of syndromes being zero, while a single error would cause an odd number of syndromes to be zero.

To construct a parity check matrix, H, for these new codes with r check bits and k data bits, first choose the $\binom{r}{1}$ combinations of one 1 and $r-1$ 0's and assign the column with a single 1 in the i^{th} place to the check bit c_i, $1 \leq i \leq r$. If $k \leq \binom{r}{3}$, choose k distinct combinations of three 1's and $(r-3)$ 0's as columns corresponding to the data bits. If $\binom{r}{5} \geq k \geq \binom{r}{3}$, choose all $\binom{r}{3}$ combinations of three 1's and $(r-3)$ 0's and $(k-\binom{r}{3})$ distinct combinations of five 1's and $(r-5)$ 0's as columns in H. If necessary, continue with 5, 7, 9... etc. bits in a column equal to 1.

For self-checking design of translators, this new code has the following important properties.

Property 1: the columns of the parity check matrix are a minimum Hamming distance two apart.

Property 2: each column of the parity check matrix is odd weight.

Property 3: if there are r check bits c_j, m bytes with parity bits p_i, and odd parity is used,

$$c_1 \oplus c_2 \oplus \cdots \oplus c_r \oplus p_1 \oplus \cdots \oplus p_m$$
$$= (r+m) \bmod 2 .$$ (See 6-Appendix)

3.2 System Implementation of the Translator

The translator consists of a syndrome generator, a single error corrector, a double error detector, a byte parity encoder, a byte parity checker and a circuit to implement the check based on Property 3 above. There are two registers, the Data Word Register, which contains data in SEC/DED code, and the Memory Data Register, containing corrected data in byte parity format.

Figure 3 shows the translator block diagram. The design of each testable block is discussed in detail in the following sections. The failure indication given by the three indicators Q, E and W are tabulated below. Each indicator is (0, 1) or (1, 0) in code space and (0, 0) or (1, 1) in error space.

Indicator	Failure Indication
$Q:(Q_0, Q_1)$	Double error in memory word or (exclusively) at least one failure in the circuit between the Data Word Register and the lines providing the indication, Q.
$E:(E_0, E_1)$	Failures in the single error corrector circuit that give a parity error in the output word from the corrector or (exclusively) a single failure in the circuit between the single error corrector and the lines providing the indication E.
$W:(W_0, W_1)$	At least single failure in the circuit between the single error corrector and the lines providing the indication W.

3.3 Syndrome Generator

The syndromes are generated in self-testing pairs $\{(s_{10}, s_{11}), (s_{20}, s_{21}), \ldots, (s_{r0}, s_{r1})\}$. In code space, each pair is either (1, 0) or (0, 1). The syndrome generators are formed from two output XOR trees, which are self-testing as discussed in Section 2.2. The XOR trees have the same number of inputs, or differ, at most, by one. In addition, the syndrome pairs $\{(s_{10}, s_{11}), (s_{20}, s_{21}), \ldots, (s_{r0}, s_{r1})\}$ take on all 2^r patterns during normal operation to ensure testability of the succeeding blocks. An algorithm for forming syndrome pairs with this property is given in the Appendix.

3.4 Double-Input Error Detection

As proven previously, with the new code and conventional implementation, single input errors change an odd number of syndromes from 1 to 0, and double input errors change an even number from 1 to 0. When the syndromes are implemented as self-testing parity check trees with dual outputs, a single input error changes an odd number of these pairs from one of $\left\{\begin{matrix}0 & 1\\ 1 & 0\end{matrix}\right\}$ to one of $\left\{\begin{matrix}0 & 0\\ 1 & 1\end{matrix}\right\}$ and a double input error changes an even number of these pairs similarly. As specified in section 2.2, a self-testing implementation of the double input error detection circuit is necessary. This is begun by using the syndrome pairs (s_{i0}, s_{i1}) as inputs to an RCCO with outputs (R_0, R_1). The RCCO output (R_0, R_1) will be in code space if all the syndromes indicate no input errors and in error space if one or more syndromes indicate an input error. Furthermore, if (R_0, R_1) feed self-testing circuits to the final error output, the syndrome generation circuits and the RCCO will be completely self-tested.

The syndrome pairs are used to generate a second signal, $P = (P_0, P_1)$ through a parity check circuit according to the equations

$P_0 = s_{10} \oplus s_{11} \oplus s_{20} \oplus s_{30} \oplus \cdots \oplus s_{r0}$ and

$P_1 = s_{21} \oplus s_{31} \oplus \cdots \oplus s_{r1}$ in order that the set $\{(P_0, P_1), (R_0, R_1)\}$ takes all four values during normal operation as proved in the Appendix. The values of (P_0, P_1) will be in code space when the syndromes indicate no input errors or an even number of input errors, and in error space when an odd number of inputs are in error. (If r is even, P_1 is inverted.)

These signals, (R_0, R_1) and (P_0, P_1), can be used to form an indication of a double error condition as follows. In this analysis, only combinations of up to two errors or failures will be considered. If (R_0, R_1) and (P_0, P_1) are in code space, then there are no single input errors or single circuit failures or any combination of single input error and single circuit failures. If (P_0, P_1) is in code space, but (R_0, R_1) is in error space, then there is either a double input error (an even number of syndromes are in error space) or a single circuit failure in generating (R_0, R_1), or a failure in (P_0, P_1) masking a single input error (which would show except for the error in (P_0, P_1)). If (P_0, P_1) is in error space but (R_0, R_1) is in code space, then there is a single circuit failure in generating (P_0, P_1) or an error in (R_0, R_1) masking a single input error. If both (R_0, R_1) and (P_0, P_1) are in error space, then there is a single input error (an odd number of syndromes are in error space).

This information can be collected in a table as follows:

Cases	(P_0, P_1)	(R_0, R_1)	Number of Combinations	Meaning
1	$\left\{\begin{matrix}0 & 1\\ 1 & 0\end{matrix}\right\}$	$\left\{\begin{matrix}1 & 0\\ 0 & 1\end{matrix}\right\}$	4	Codespace
2	$\left\{\begin{matrix}0 & 1\\ 1 & 0\end{matrix}\right\}$	$\left\{\begin{matrix}0 & 0\\ 1 & 1\end{matrix}\right\}$	4	Double Input Data Error or Circuit Error
3	$\left\{\begin{matrix}0 & 0\\ 1 & 1\end{matrix}\right\}$	$\left\{\begin{matrix}0 & 1\\ 1 & 0\end{matrix}\right\}$	4	Circuit Error
4	$\left\{\begin{matrix}0 & 0\\ 1 & 1\end{matrix}\right\}$	$\left\{\begin{matrix}0 & 0\\ 1 & 1\end{matrix}\right\}$	4	Single Input Data Error

If

$$Q_0 = P_0 \oplus R_0$$

$$Q_1 = \overline{P_1 \oplus R_1}$$

then (Q_0, Q_1) are in $\left\{\begin{matrix}0 & 1\\ 1 & 0\end{matrix}\right\}$ for cases 1 and 4 from the above table and in $\left\{\begin{matrix}0 & 0\\ 1 & 1\end{matrix}\right\}$ for cases 2 and 3. Moreover, the circuit (Q_0, Q_1) is self-testing and has testable inputs. Thus, the output pair (Q_0, Q_1) signals a double input data error and is self-testing, so the circuits which generate the dual output syndromes, the RCCO with outputs (R_0, R_1), and the parity check (P_0, P_1) are self-testing as specified.

3.5 Single Error Correction

For the correction of single errors, to save circuitry, reduce the self-testing syndrome pairs to single lines. Set $S_i = s_{i0} \oplus s_{i1}$, $i = 1,\ldots,r$ and independently generate $\bar{S}_i = \overline{(s_{i0} \oplus s_{i1})}$, $i = 1,\ldots,r$. The independent generation of $\bar{S}_i$ makes them self-checking when there are 0 or 1 input data errors, and makes them self-testing over either code space or single error space. These statements are proved in the Appendix.

The data or check bit correction circuit is formed as the XOR of the data or check bit and the conjunct of syndromes formed by examining the column corresponding to the data or check bit in the parity check matrix. If the i^{th} bit in this column is 0, S_i appears unnegated (for odd parity) in the conjunct and, if the bit is 1, S_i appears negated. So, if the j^{th} column of a four row parity, corresponding to d_j is given as the transpose of (1110), the d_j is expressed in its corrected form, d_{jc}, by $d_{jc} = d_j + \bar{S}_1\bar{S}_2\bar{S}_3S_4$, where $\bar{S}_1\bar{S}_2\bar{S}_3S_4$ is the corrective conjunct of syndromes. In odd parity, when the data and check bits are correct, all S_i are 1 and no corrections are made. This means that the AND gates, forming the corrective conjuncts, are tested for stuck-at-one only when there are no input data errors.

After the data is corrected, each byte is fed into a parity generation circuit and the byte with generated parity is put into the Memory Data Register (MDR).

3.6 Final Output Signals

Three distinct sets of check signals are generated as outputs of the translator: self-testing line pairs $Q:(Q_0, Q_1)$, $W:(W_0, W_1)$ and $E:(E_0, E_1)$. The generation of Q has been explained in Section 3.4. The data in the MDR is parity checked (to check the parity generation circuits as well as data to be written into the memory). The dual outputs of these self-testing parity checkers are fed as inputs in an RCCO circuit which has outputs (W_0, W_1).

To implement the check of Property 3 of the code, the generated parity bits are fed into a single output parity check circuit with output E_0 . The check bits are fed into another parity check circuit with output E_1 (possibly inverted, depending upon the value of (r+m) mod 2). These parity trees are self-checking. The pair (E_0, E_1) is in code space if the data bits $d_1,\ldots,d_k$ and the check bits $c_1,\ldots,c_r$ are correct, as is shown in the Appendix. It is much more economical, in terms of gates saved, to implement this final monitor circuit than to perform the correction using self-testable syndrome pairs and double line circuits for all data and check bit correction functions.

If desired, the signal pairs (Q_0, Q_1), (W_0, W_1) and (E_0, E_1) can be reduced to a single error signal pair by an RCCO. However, for diagnostic purposes, separating the types of errors is desirable, so this reduction can be dictated by the strategy of recovery planning. All self-checking and self-testing features would remain independent of this final RCCO circuit.

3.7 The Memory WRITE Cycle

The checked word is taken from the byte parity register in byte parity-encoded form and re-encoded into SEC/DED encoded form by the same self-testing translator. The check bits, c_i, are generated from the data bits in the incoming word as prescribed by the parity check matrix. The same set of data bits that were combined through XOR trees in the READ cycle to form S_i are now used in the WRITE cycle to form the c_i so that literally the same circuitry can be used for both cycles.

The generated c_i can now be checked by the p_i from the byte parity MDR by using the same circuitry with output (E_0, E_1) which was used to check the output used in the READ cycle.

4. RESULTS AND CIRCUIT COMPARISONS

The SEC/DED translator herein described has the following properties:

1) All circuits which signal the occurrence of double input data errors and/or single circuit failures are completely self-testing when there are no input data errors or when there are single input data errors.

2) All single failures in circuits which correct single input data errors are detected during normal operation (i.e., code space) except for the correction AND_i gate output lines stuck-at-0 (s-a-0), the negated syndrome lines $(\bar{S}_i)$ s-a-0, and the syndrome lines (S_j) stuck-at-1 (s-a-1) and s-a-0. The above mentioned failures which are undetected will not produce erroneous outputs.

3) If there are single input errors, any of the following single failures resulting in lines stuck-at-0 and not detected previously are detected: the AND_i gate output lines, the negated syndrome lines $(\bar{S}_i)$ and the syndrome lines S_i. The syndrome line (S_j) s-a-1 is not detected, but this will not produce an erroneous output.

4) All double circuit failures, one of which is a syndrome line (S_j) s-a-1, are detected in code or single error space and will not produce erroneous output when not detected.

5) If the syndrome line S_j is s-a-1 and an input double error occurs, the input double error is signaled and the circuit failure may be detected. In the event it is not detected, erroneous output is produced.

Finally, as an ultimate test of the error correcting and detecting capabilities of the translator, assume that multiple errors (two or more) are induced by multiple failures occurring in a single memory cycle. If the errors are of the small percentage which may be undetected when a particular data pattern is being read, then undetected errors can occur for a few memory cycles until the data pattern changes so the error is detected.

The extra cost in circuitry used for a self-testable translator is shown in Table 1 .

The implementation of SEC/DED code translators with self-checking and self-testing logic circuitry can be achieved rather economically in terms of extra circuitry: an effective doubling of the word sizes in Table 1 produced more than a factor of two decrease in the percentage of extra circuitry required. The reason for this is that the syndrome generation, the correction circuitry, byte parity generation, and byte parity checking are all unchanged, whether the circuit is self-checking or not. The remaining circuits, the ones which detect a change in one or more syndromes, the parity generator across the syndrome outputs, the double error signal, and the final error check on the memory data register have been made self-testing. The number of inputs to these circuits depends upon r and the number of bytes, and so are relatively small. Thus, proper application of self-testing circuit design techniques enabled excellent results to be obtained.

5. CONCLUSIONS

Error detection and correction circuitry is used to increase memory system reliability by minimizing the possibility of continued machine processing in the presence of undetected memory errors. New memory organizations, with random parallel access, are explicitly structured to localize component failures so that these failures are normally manifested as single-bit errors. Fault toleration and attendant increased reliability is then gained by using Hamming SEC/DED codes. However, in conventional systems, the probability of component failure is still not negligible, especially in encoding and decoding circuitry, and these may now result in undetected double input errors and/or erroneously corrected data.

Dynamically-tested decoding circuitry, implemented with the function of correcting erroneous data without mistaking data errors for errors caused by circuit faults, and vice-versa, has been designed. Methods have been delineated for designing SEC/DED to byte parity translators which

1) correct all single input data errors,
2) detect all double input data errors,
3) under normal conditions, will never erroneously correct input data nor not detect double errors,
4) signal the existence of any component failure in the decoding circuitry,
5) have a small percentage of circuitry more than that required by conventional translators.

6. APPENDIX-MATHEMATICAL PROOFS

In this appendix, the five properties and statements used in designing the self-testing translator are proved to be true. All the self-testing and self-checking capabilities of the circuits described in the text rest upon these proofs. The design of such units cannot be understood apart from an understanding of these proofs.

1) Property 3 of the new code states that

$$c_1 \oplus \cdots \oplus c_r \oplus p_1 \oplus \cdots \oplus p_m = (r+m) \bmod 2 .$$

The XOR of the syndromes is $r \bmod 2$. If the syndrome equations are XOR'd, then from property 2 of the parity check matrix, each c_j appears once and each d_i an odd number of times, so

$$d_1 \oplus d_2 \oplus \cdots \oplus d_k \oplus c_1 \oplus \cdots \oplus c_r = r \bmod 2.$$

Forming the XOR of all m bytes,

$$d_1 \oplus d_2 \oplus \cdots \oplus d_k \oplus p_1 \oplus \cdots \oplus p_m = m \bmod 2.$$

The final result follows by XORing these two equations.

2) Let $\rho = \min_i (\rho_i)$, where ρ_i is the number of 1's in the i^{th} row of the parity check matrix.

The following algorithm will select subsets of the set $D_i = \{d_{i_1}, \ldots, d_{i_{\rho_i - 1}}, r_{\rho_i}\}$ of bits that have 1's in the i^{th} row of the parity check matrix so that $\{(s_{10}, s_{11}), (s_{20}, s_{21}), \ldots, (s_{r0}, s_{r1})\}$ will take on all 2^r values, and the syndrome generator will have minimum circuit delay.

Step 1. Choose any of $\lceil \rho/2 \rceil$ elements of D_1 and form s_{10} as an XOR of these elements.

Step 2. Choose $\lceil \rho/2 \rceil$ elements of D_2 with one element different from the set chosen in step 1, and form s_{20} as XOR of these elements.

Step r. Choose $\lceil \rho/2 \rceil$ elements of D_r with one element different from the elements of the union of sets chosen in step 1, step 2,..., step (r-1) and form s_{r0} as XOR of these elements.

If the required choice at any stage cannot be made, repeat the process from step 1 again choosing in each step $\lceil \rho/2 \rceil -1$ elements. If again unsuccessful, repeat the process choosing $\lceil \rho/2 \rceil -2$ elements, and so on. Note that the ultimate choice of a single element in each step will always work. Since the r sets are linearly independent, the S_{i0}'s take on all 2^r values and hence so will $\{(s_{10}, s_{11}), (s_{20}, s_{21}),\ldots, (s_{r0}, s_{r1})\}$. Choosing the number of bits involved in the generation of s_{i0} as near to $\lceil \rho/2 \rceil$ as possible gives minimum circuit delay.

3) In Figure 3, the set $\{R_0, R_1), (P_0, P_1)\}$ takes on all four values $\{(0,1), (0,1)\}$, $\{(0,1), (1,0)\}$, $\{(1,0), (0,1)\}$ and $\{(1,0), (1,0)\}$.

In section 3.4, it was mentioned that

$$P_0 = s_{10} \oplus s_{11} \oplus s_{20} \oplus \cdots \oplus s_{r0}$$

and

$$R_0 = C_1 s_{10} \vee C_2 s_{11}$$

where C_1 and C_2 are Boolean functions independent of s_{10} and s_{11} and $C_1 \oplus C_2 = 1$ during normal operation.

Choose values of s_{i0} which make $P_0 = 0$, and $R_0 = \alpha$ (α is either 0 or 1). Change the value of S_{10}. Now, $P_0 = 0$, $R_0 = \bar{\alpha}$. Repeat the argument with $P_0 = 1$.

4) Testability of the single-error-corrector circuit lines.

The implementation equations in the single-error-corrector are

$$S_i = S_{i0} \oplus S_{i1} \qquad i = 1,2,\ldots,r$$

$$\bar{S}_i = \overline{S_{i0} \oplus S_{i1}} \qquad i = 1,2,\ldots,r$$

$$d_{ic} = d_i \oplus \text{AND}_i(T) \qquad i = 1,2,\ldots,r$$

$$c_{jc} = c_j \oplus \text{AND}_j(T) \qquad j = 1,2,\ldots,r$$

where AND_i is the AND gate into which T feeds, and T is an r-dimensional vector with an odd number of elements from $\{\bar{S}_1, \bar{S}_2,\ldots,\bar{S}_r\}$ and its remaining elements from $\{S_1, S_2,\ldots,S_r\}$ as determined by a column in the parity check matrix. Also, S_i is one of the elements iff $\bar{S}_i$ is not. For example, $\bar{S}_1\bar{S}_2\bar{S}_3S_4$ and $S_1\bar{S}_2S_3S_4$ would be valid $\text{AND}_i(T)$'s for $r = 4$.

The investigation of testability of the lines S_i, $\bar{S}_i$, AND_i $i = 1,2,\ldots,(k+r)$ is carried out as follows. At first testability is examined in code space. Then, testability in single error space is examined with concentration on those failures which were undetectable in code space. Having obtained the set of single failures which are not detected either in code space or single error space, it is determined if further accumulation of undetected failures is possible.

A failure in the single error corrector circuit is detectable if the byte parity bits generated from the corrected data bits and the corrected check bits do not satisfy Property 3.

Testability in Code Space

In code space S_i, $\bar{S}_i$ and AND_i never take on the value 0, 1 and 1 respectively and hence the failures S_i stuck-at-1 (s-a-1), $\bar{S}_i$ stuck-at-0 (s-a-0) and AND_i stuck-at-0 cannot be detected.

S_i s-a-0: For the error to originate, it is required that S_i should actually be 1, so $\bar{S}_i = 0$ and no correction should be made. According to the property of T cited above, each correction conjunct $\text{AND}_j(T)$ must either contain S_i or $\bar{S}_i$. None of the AND gates are activated and hence no erroneous correction results. The failure is undetected, and no error is introduced.

$\bar{S}_i$ s-a-1: The AND gate corresponding to the correction conjunct $S_1S_2 \cdots S_{i-1} \bar{S}_iS_{i+1} \cdots S_r$ is erroneously activated causing a check-bit to be erroneously corrected, so the check output E_1 is wrong. Since all data bits are correct, the output E_0 is unchanged, and the pair (E_0, E_1) signal an error.

AND_i s-a-1: The said AND gate is erroneously activated so one data bit and hence one byte parity bit is wrong or (exclusively) one check bit is wrong, and in either case the pair (E_0, E_1) will signal an error.

Hence, the failures undetected in code space are S_i s-a-0, S_i s-a-1, $\bar{S}_i$ s-a-0 and AND_i s-a-0.

Testability in Single Error Space

From Property 1 of the new code, over single error space a single error in S_i or $\bar{S}_j$ will never result in an erroneous correction since a change of one bit will not change one column of the parity check matrix into another.

Consider the failures undetected in code space.

S_i s-a-0. For an error to originate, S_i=1, S_i s-a-0 and $\bar{S}_i = 0$ will inactivate all AND gates and no correction is made. One bit of the presumably corrected word is in error, and this will be detected by the check pair (E_0, E_1).

$\bar{S}_i$ s-a-0: For an error to originate, $\bar{S}_i = 1$ so $S_i = 0$. The proof is similar to that for S_i s-a-0 and the error is detected.

AND_i s-a-0: When the single error pattern that should activate this AND gate occurs, no correction will be made. The above proofs show this error will be detected.

S_i s-a-1: For an error to originate, $S_i = 0$, so $\bar{S}_i = 1$. Such a pattern will be one that activates an AND gate corresponding to a T that contains $\bar{S}_i$ and hence not S_i. The error thus does not affect this AND gate and the right correction is made. No erroneous correction is made and the failure is undetected. The output word is correct.

The only single failure that is undetected both in code space and single error space is S_i s-a-1, so failures in S_i s-a-1 can accumulate.

5) Testability of various lines of the single-error-corrector given that one S_j is s-a-1.

For testability in code space, the arguments follow through just as before with the same result viz. undetected failures are $\bar{S}_i(i \neq j)$ s-a-0, $S_i(i \neq j)$ s-a-1, $\bar{S}_i$ s-a-0 and AND_i s-a-0.

For testability in single error space, the arguments follow through just as before for S_i s-a-0, $\bar{S}_i$ s-a-0, and AND_i s-a-0.

S_i $(i \neq j)$ s-a-1: The single error pattern that can cause errors to originate at both sites of failures must have $S_i = S_j = 0$, so $\bar{S}_i = 1$ and $\bar{S}_j = 1$. The AND gate that should be activated by this pattern must correspond to a T_1 which contains $\bar{S}_i$ and $\bar{S}_j$ and hence not S_i or S_j and hence the right correction is made. Write $T_1 = \{\bar{S}_i, \bar{S}_j\} \cup \mathcal{T}$. Then $\mathcal{T}$ must contain an odd number of elements from $\{\bar{S}_1, \bar{S}_2, \ldots, \bar{S}_r\} - \{\bar{S}_i, \bar{S}_j\}$ from the code property 2. If there are AND gates corresponding to $T_2 = \{\bar{S}_i, S_j\} \cup \mathcal{T}$, $T_3 = \{S_i, \bar{S}_j\} \cup \mathcal{T}$ and $T_4 = \{S_i, S_j\} \cup \mathcal{T}$, then other data or check bits will be changed. From Property 2, again, the AND gate corresponding to T_4 is the only one which exists, so that another output bit is erroneously changed, the XOR of the parity bits of the MDR is erroneous, and the failure is detected.

Thus, accumulation of failures beyond a single S_i s-a-1 will never go undetected in code and single error space

REFERENCES

1. Allen, C. A., "Design of Digital Memories that Tolerate All Classes of Defects", Technical Report No. AFAL-TR-66-223 or SU-SEL-66-031, prepared under USAF Contract AF 33(657)-11586, Stanford Electronics Laboratories, Stanford, California, 1966.
2. Carter, W. C. and Schneider, P. R., "Design of Dynamically Checked Computers", IFIPS 1968 Edinburgh, Scotland, August, 1968.
3. Bouricius, W. G., Carter, W. C., Duke, K. A., Roth, J. P., and Schneider, P. R., "Interacting Design of Self-testing Circuitry", Purdue Centennial Year Symposium, April, 1969.
4. Wadia, A. B., "Investigations into the Design of Dynamically Checked Arithmetic Units", Ph. D. Thesis, Division of Engineering and Applied Physics, Harvard University, Cambridge, Massachusetts, January 1970.
5. Sellers, F. F., Hsiao, M. and Bearnson, L. W., "Error Detecting Logic for Digital Computers", McGraw Hill Book Company, New York, 1968.
6. Hamming. R. W., "Error Detecting and Error Correcting Codes", Bell System Tech. J. 29, 1950, 147-160.
7. Henle, R.A., "Array Logic, Technical Note No. 8", Private Communication, IBM Components Division, February 27, 1969.

TABLE 1

n	k	r	Conventional SEC/DED to Byte Parity Circuits	Self-Checking SEC/DED to Byte Parity Circuits	% Increase
32	26	6	1050	1130	7.4
64	57	7	2300	2380	3.5
128	120	8	4950	5000	1.1

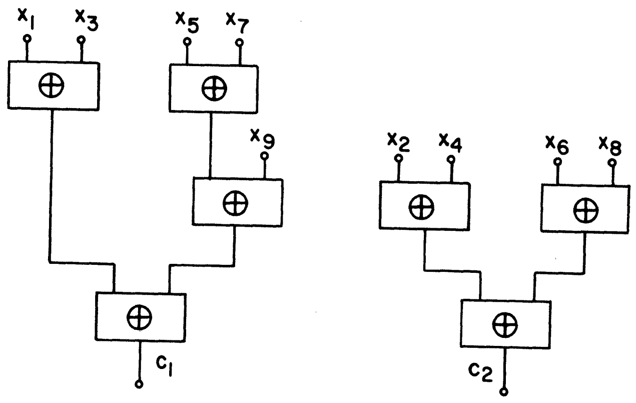
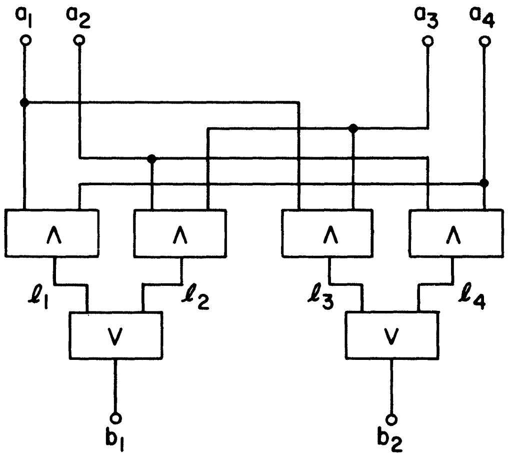
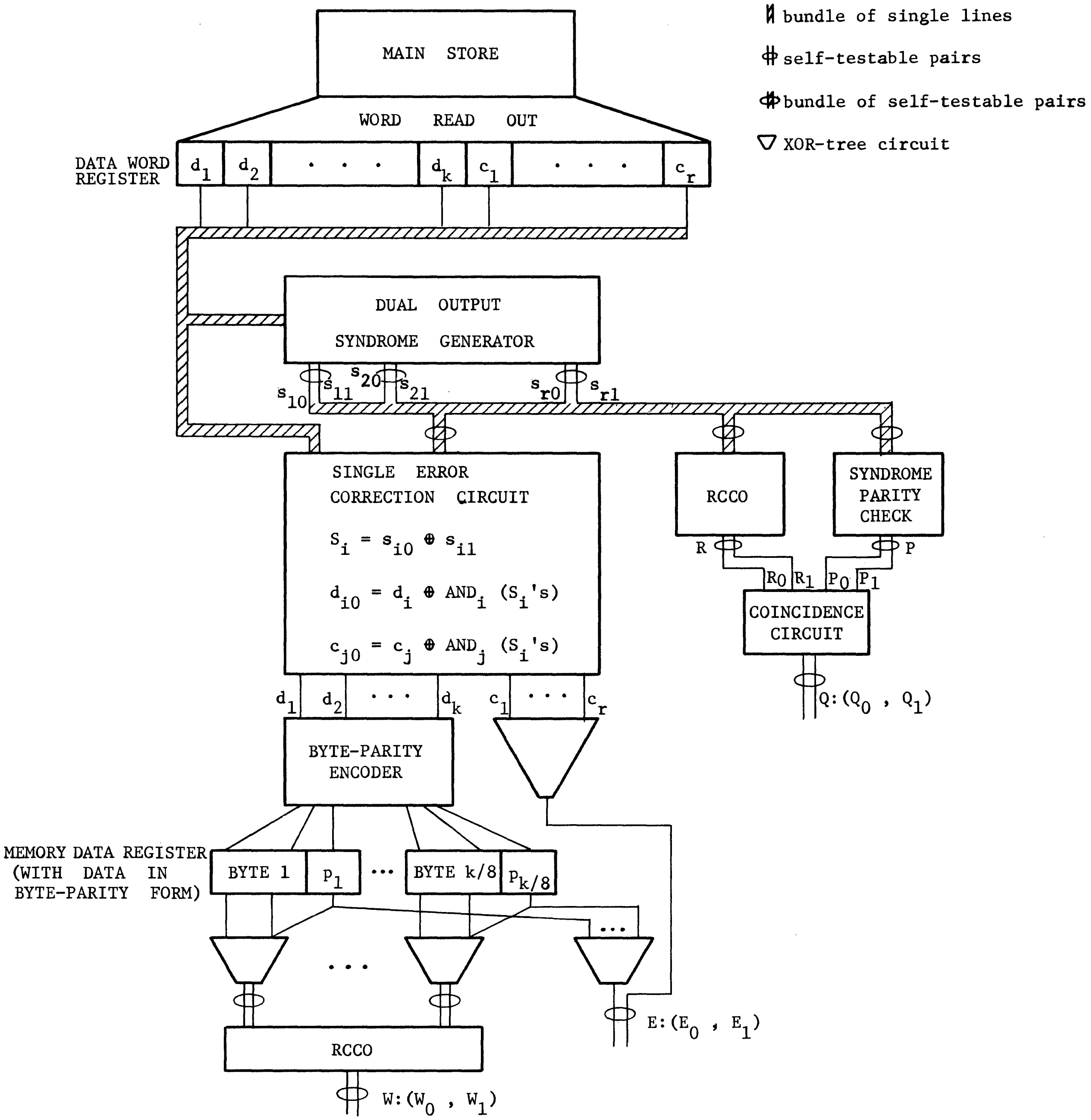

x_1
x_3
x_5
x_7
x_9
x_2
x_4
x_6
x_8
⊕
c_1
c_2

FIGURE 1. PARITY CHECKER/SYNDROME GENERATOR


a_1
a_2
a_3
a_4
∧
ℓ_1
ℓ_2
ℓ_3
ℓ_4
∨
b_1
b_2

a_1	a_2	a_3	a_4	ℓ_1	ℓ_2	ℓ_3	ℓ_4	b_1	b_2
0	1	0	1	0	0	0	1	0	1
0	1	1	0	0	1	0	0	1	0
1	0	0	1	1	0	0	0	1	0
1	0	1	0	0	0	1	0	0	1
0	0	x	x	0	0	0	0	0	0
x	x	0	0	0	0	0	0	0	0
1	1	1	x	x	1	1	x	1	1
1	1	x	1	1	x	x	1	1	1
1	x	1	1	1	x	1	x	1	1
x	1	1	1	x	1	x	1	1	1

FIGURE 2. REDUCTION CIRCUIT FOR CHECKER OUTPUTS



bundle of single lines
self-testable pairs
bundle of self-testable pairs
XOR-tree circuit
MAIN STORE
WORD READ OUT
DATA WORD REGISTER
d_1
d_2
d_k
c_1
c_r
DUAL OUTPUT SYNDROME GENERATOR
s_{10}
s_{11}
s_{20}
s_{21}
s_{r0}
s_{r1}
SINGLE ERROR CORRECTION CIRCUIT
$S_i = s_{i0} \oplus s_{i1}$
$d_{i0} = d_i \oplus AND_i\ (S_i\text{'s})$
$c_{j0} = c_j \oplus AND_j\ (S_i\text{'s})$
RCCO
SYNDROME PARITY CHECK
R
P
R_0
R_1
P_0
P_1
COINCIDENCE CIRCUIT
$Q{:}(Q_0\ ,\ Q_1)$
BYTE-PARITY ENCODER
MEMORY DATA REGISTER (WITH DATA IN BYTE-PARITY FORM)
BYTE 1
P_1
BYTE k/8
$P_{k/8}$
RCCO
$W{:}(W_0\ ,\ W_1)$
$E{:}(E_0\ ,\ E_1)$

FIG. 3 MEMORY TRANSLATOR CIRCUIT IN READ CYCLE

A METHOD OF DERIVING MINIMAL COMPLETE SETS
OF TEST-INPUT SEQUENCES USING BOOLEAN DIFFERENCES

P. N. Marinos, Associate Professor, Department of Electrical Engineering
Duke University, Durham, N. C. 27706

ABSTRACT

This paper deals with a fault-detection and diagnosis technique based on Boolean differences. A brief review of the notion of a Boolean difference is presented, and an algorithm for obtaining minimal, complete sets of test-input sequences based on the Boolean differences of a switching function is formulated. Illustrations demonstrating the use of the technique are presented.

INTRODUCTION

One of the requirements of a design process is that the final product be functionally sound and capable of maintaining its functional integrity over a specified period of time with some degree of certainty. This requirement becomes even more important when the function to be performed is of a critical nature. It is therefore advisable that the user of a system have at his disposal some means of determining system integrity through an appropriate checking scheme which may be applied either periodically or on a continuous basis.

Error detection and error correction in switching systems may be carried out either through hardware or through the use of software redundancy. Many software and hardware redundancy schemes have been proposed for error control, and the reader can easily locate published accounts of the various techniques suggested in the open literature [1-5].

One of the error control schemes which is often used in fault detection and fault diagnosis involves the application of a known test sequence to a discrete parameter system, and subsequent observation of the system response (output sequence). The idea here being that if the system is functionally sound its response should be identical to the system's normally expected behavior due to that particular input sequence. The system response will, in general, depend on the applied input and the state of the system. Even though the type of system involved does dictate the approach to be followed in performing the fault diagnosis tests, the method of obtaining the appropriate test input sequences is the same for both memoryless systems as well as those endowed with memory properties.

The choice of test-input sequences will, in general, depend on the type of fault diagnosis one wishes to perform. This may cover, of course, a rather wide spectrum of choices varying from a simple test of applying the "most likely to occur input state" and observing the system response, to the application of an elaborate input sequence for diagnosing any single (or multiple) failures within the system. The approach to be followed here is based on the notion of the Boolean difference, an analytical tool which has been previously employed in error detection and fault diagnosis [5-8].

BOOLEAN DIFFERENCES

The Boolean difference of a switching function $F(x_1,x_2,\ldots,x_n)$, is defined with respect to any one of its variables as follows[5].

$$D_i F = F(x_1,x_2,\ldots,x_i,\ldots,x_n) \oplus F(x_1,x_2,\ldots,\bar{x}_i,\ldots,x_n) \qquad (1)$$

where $\oplus$ denotes the exclusive-OR operation, and D_iF represents the Boolean difference of the switching function F with respect to x_i. One of the properties of the Boolean difference, D_iF, is that $D_iF \equiv 0$ when F is unconditionally independent of x_i; and $D_iF \equiv 1$ when F is unconditionally dependent on x_i. In the case in which D_iF is itself another switching function, we will say that the dependence of F on the particular variable x_i is conditional. These properties are easily verified by noting that $D_iF \equiv 0$ implies that both forms of F in Eq. (1) are identical; that is to say, the state of x_i is immaterial as far as the behavior of F is concerned. Similarly, $D_iF \equiv 1$ implies that the two forms of F in Eq. (1) are complementary or, equivalently, that function F does depend on the state of the variable x_i. The third and most important possibility is for D_iF to be a switching function. If this be the case, D_iF assumes a functional form in which the literal x_i is missing. This is easily established by noting that due to the binary nature of x_i one can rewrite Eq. (1) in the form,

$$D_iF = F(x_1,x_2,\ldots,0,\ldots,x_n) \oplus F(x_1,x_2,\ldots,1,\ldots,x_n) \qquad (2)$$

which if expanded yields the functional

$$D_iF = F_i(x_1,x_2,\ldots,x_{i-1},x_{i+1},\ldots,x_n) \qquad (3)$$

Therefore, one may obtain the Boolean difference of a function with respect to a variable x_i by replacing x_i in the function by 0 and 1 and exclusive-ORing the two forms resulting from the substitutions. For the sake of brevity, let

$$F_i^{\circ} = F(x_1,x_2,\ldots,0,x_{i+1},\ldots,x_n)$$

Part of this work was carried out while the author was with Bell Telephone Laboratories, Inc. Greensboro, N. C. (summer, 1969).

and

$$F_i^1 = F(x_1,x_2,\ldots,1,x_{i+1},\ldots,x_n) \qquad (4)$$

In view of Eqs.(4), one may rewrite (2) as follows:

$$D_iF = F_i^{\circ} \oplus F_i^1 \qquad (5)$$

The following example illustrates the manner in which single Boolean differences (Boolean differences with respect to one variable) are obtained.

Example 1: Determine the Boolean difference of the switching function

$F(x_1,x_2,x_3) = x_1\, x_2 + x_1\, \bar{x}_3$ with respect to x_1.

Applying expressions (4) given above, one has

$F_1^{\circ} = 0$ and $F_1^1 = x_2 + \bar{x}_3$

which in view of expression (5) yield

$D_1F = 0 \oplus (x_2 + \bar{x}_3) = x_2 + \bar{x}_3$

Thus, the Boolean difference in this case is a switching function in which x_1 is missing. This function when equal to 1 ($D_1F = 1$) will generate the conditions under which F will depend on x_1; and when equal to 0, it will generate those conditions under which F will be independent of x_1. It is therefore obvious that F is dependent on x_1 when $x_2 = 1$ or $x_3 = 0$ (or both); similarly, F is independent of x_1 when $x_2 = 0$ and $x_3 = 1$. This type of dependence (independence) of F on x_1 is known as a conditional dependence (independence).

The concept of Boolean difference may be cast in various forms some of which are presented below without proof.

$$D_i\bar{F} = D_iF \quad (\bar{F} = \text{complement of } F) \qquad (6)$$

$$D_i[D_jF] = D_j[D_iF] \qquad (7)$$

$$D_i[F\cdot G] = F\cdot D_iG \oplus G\cdot D_iF \oplus D_iF\cdot D_iG \qquad (8)$$

$$D_i[F+G] = \bar{F}\cdot D_iG \oplus \bar{G}\cdot D_iF \oplus D_iF\cdot D_iG \qquad (9)$$

$$D_i[F \oplus G] = D_iF \oplus D_iG \qquad (10)$$

For cases in which one is interested in finding the dependence of a switching function on two variables, x_i and x_j, simultaneously, the so-called double Boolean difference is expressible in the form

$$D_{ij}F = F_{ij}^{\circ\circ} \oplus F_{ij}^{11} \qquad (11)$$

where $F_{ij}^{\circ\circ}$ and F_{ij}^{11} are the valuations of the switching function F at $x_i = x_j = 0$ and $x_i = x_j = 1$, respectively.

If one is interested in determining the dependence of F on either x_i or x_j, but not both, the appropriate form to be used is

$$D_{i\oplus j}F = D_iF + D_jF \qquad (12)$$

Finally, the appropriate formulation which will provide the dependence of F on either x_i or x_j or both x_i and x_j simultaneously is given by

$$D_{i+j}F = D_{ij}F + D_{i\oplus j}F \qquad (13)$$

Although expressions (11) through (13) provide the various forms of double Boolean differences, the concept can be extended to any number of variables.

BOOLEAN DIFFERENCES AND FAULT DETECTION IN SWITCHING SYSTEMS.

The Boolean difference of a switching function F, with respect to one of its variables, x_i, may be thought of as being the analytical representation of the conditions under which the switching function becomes dependent on the value of that particular variable, x_i. These conditions are expressed by the code groups which satisfy the equation $D_iF=1$.

Prior to developing the algorithm used for deriving the desired test-input sequences from the Boolean differences of a function, a few remarks concerning the type of faults considered here are in order.

It is assumed that the faults likely to occur in the system under consideration are of the "stuck-at-zero" (s-a-o) and "stuck-at-one" (s-a-1) type. It is further assumed that faults may occur at the input lines as well as at the internal nodes of the system. Because of the fact that internal-node faults are dependent on the particular hardware implementation used, they will be treated separately.

(a) Input-line faults: The effect of input-line faults is to make the switching function describing a system independent of the variables which have been faulted (s-a-o or s-a-1). Thus, if F_a is the correct functional description of a system, and F_b is the function to which F_a is transformed as a result of faults at the input lines, then one must have that

$$D_iF_a \not\equiv 0 \text{ and } D_iF_b \equiv 0$$

for all input lines (variables) x_i which have been faulted. The above relations imply that the Boolean differences of F_a could be used to detect faults of this type, and the following example clearly illustrates the point.

Example 2: Investigate the logical dependence of the switching function

$$F_a(x_1,x_2,x_3) = x_1\bar{x}_2 + x_2x_3 \qquad (14)$$

on the variables x_1, x_2 and x_3, each being considered separately.

The Boolean differences of F_a are:

$$\begin{aligned} D_1F_a &= \bar{x}_2 \\ D_2F_a &= x_1\bar{x}_3 + \bar{x}_1x_3 \\ D_3F_a &= x_2 \end{aligned} \qquad (15)$$

The Boolean differences expressed by (15) represent the conditions of dependence of F_a on its three variables x_1, x_2 and x_3. For instance, $D_1F_a = \bar{x}_2$ implies that F_a will depend on the value of x_1 if, and only if, $x_2 = 0$. Therefore, one may now use this information to check the system with respect to x_1 by applying sequentially the test-input code groups

$$x_1x_2x_3 = 00d \text{ and } x_1x_2x_3 = 10d \qquad (16)$$

where d denotes a "don't care" value for a variable. It should be noted that in the code groups selected x_2 is maintained at state "0" (condition of dependence) while the variable x_1 (variable with respect to which the Boolean difference is obtained) is caused to change state (value). If input line x_1 is functionally sound (fault-free) then F_a should change state under test-input sequence (16). This test is quite simple to carry out, but the information it yields is very significant since it will cause no change in the state of F_a when x_1 is (s-a-o) or (s-a-1); a (s-a-o) fault will transform F_a into $F_{bo} = x_2x_3$ while a (s-a-1) will yield $F_{b1} = \bar{x}_2 + x_3$ (both functions being independent of x_1). Similar tests must be carried out with respect to x_2 and x_3. In the case of x_2, the conditions under which a change in x_2 should cause a change in the state of F_a are

$$x_1x_3 = 10 \qquad \text{and} \qquad x_1x_3 = 01$$

The resulting test-input sequences from these conditions are

$$\begin{aligned} x_1x_2x_3 &= 100 \text{ — } 110 \\ x_1x_2x_3 &= 001 \text{ — } 011 \end{aligned} \tag{17}$$

In a similar manner, one can verify that

$$x_1x_2x_3 = d10 \text{ — } d11 \tag{18}$$

is the test-input sequence associated with variable x_3.

The above results may be summarized as follows:

The status of the input lines in a switching system is conveniently checked for (s-a-o) and (s-a-1) faults by:

(i) obtaining the single Boolean differences of the system-describing switching function;

(ii) identifying the conditions under which the function will depend on each of its variables; and

(iii) forming the test-input sequences to be used with each of the variables of the function.

Steps (i) and (ii) above are straight-forward but step (iii) requires further clarification and will be treated separately later.

(b) Internal-node faults: With respect to internal faults, the Boolean difference by itself may not yield test-input sequences capable of checking completely all the internal nodes of a system for possible (s-a-o) or (s-a-1) faults. Before we examine the circumstances under which this may occur, we must define the terms: (i) Embedded minterm and (ii) Companion set.

Embedded minterm: Consider the function F_a of $n > 1$ non-redundant variables expressed in its canonical minterm form. We will say that a minterm $m_q^{(n)}$ of F_a is an embedded one if F_a contains, in addition to $m_q^{(n)}$, all the minterms generated from $m_q^{(n)}$ by complementing (one at a time) each of the literals in $m_q^{(n)}$. For instance, if $m_q^{(3)} = x_1\bar{x}_2\bar{x}_3$, in order for $m_q^{(3)}$ to qualify as an embedded minterm of a 3-variable function F_a, one must have that $\bar{x}_1\bar{x}_2\bar{x}_3$, $x_1x_2\bar{x}_3$, and $x_1\bar{x}_2x_3$ are also minterms contained in F_a. This collection of minterms are called the adjacencies of $m_q^{(n)}$ and form a set to be known as the companion set of $m_q^{(n)}$.

One of the most important characteristics of embedded minterms is that when a fault eliminates only such terms in a given switching function F_a, the resulting function F_b is such that its Boolean differences cover those of F_a; i.e., $D_iF_b \supset D_iF_a$ for all inputs x_i. Therefore, a test-input sequence based on the Boolean differences of F_a will be insufficient for detecting those faults causing the elimination of only embedded minterms. The validity of this assertion is established in the proof of Theorem 1.

Theorem 1: Let F_a be a n-variable switching function with embedded minterms. Show that the occurrence of faults which cause only the elimination of such minterms will transform F_a to a new function F_b such that $D_iF_a \subset D_iF_b$, where $i = 1,2,\ldots,n$.

Proof: Let $m_q^{(n)}$ represent an embedded minterm in F_a. Evaluation of the Boolean difference of F_a with respect to any of its variables x_i will result in F_{ai}° and F_{ai}^{1} such that

$$F_{ai}^{\circ} = \dot{x}_i\, m_q^{(n-1)} + \text{Other terms}$$

and

$$F_{ai}^{1} = \bar{\dot{x}}_i\, m_q^{(n-1)} + \text{Other terms}$$

where $\dot{x}_i$ denotes "either x_i or its complement, $\bar{x}_i$," and $m_q^{(n-1)}$ represents the minterm resulting from the remaining $(n-1)$ variables $(x_1,\ldots,x_{i-1},x_{i+1},\ldots x_n)$. The above relationships follow from the fact that $x_i\, m_q^{(n-1)}$ and $\bar{x}_i\, m_q^{(n-1)}$ are elements of the companion set of $m_q^{(n)}$.

Using expression (5), one has

$$\begin{aligned} D_iF_a &= F_{ai}^{\circ} \oplus F_{ai}^{1} \\ &= (m_q^{(n-1)} + A) \oplus (m_q^{(n-1)} + B) \end{aligned}$$

where A and B represent "other terms" in F_{ai}° and F_{ai}^{1}, respectively, If $m_q^{(n)} = x_i\, m_q^{(n-1)}$, a fault eliminating $m_q^{(n)}$ would result in a function F_b such that

$$D_iF_b = (m_q^{(n-1)} + A) \oplus B.$$

Since

$$D_iF_a = \bar{m}_q^{(n-1)}(\bar{A}\,B + A\,\bar{B})$$

and

$$D_iF_b = \bar{m}_q^{(n-1)}(\bar{A}\,B + A\,\bar{B}) + m_q^{(n-1)}\bar{B}$$

it becomes evident that $D_iF_a \subset D_iF_b$, where $i = 1,2,\ldots n$. The proof is identically the same when two or more embedded minterms are faulted simultaneously. Q.E.D.

Theorem 1 establishes the fact that the Boolean difference of a function F_a will not provide satisfactory test-input sequences for checking faults which affect only the embedded minterms that might exist in the switching function F_a. Although the occurrence of a fault (or faults) affecting only possible embedded minterms without spreading to others is a highly unlikely eventuality in an actual system, their possible presence, nevertheless, must be taken into account when deriving test-input sequences with the objective of exercising completely the internal nodes of a system. This point is illustrated clearly by the function of Example 3.

Example 3: Show that the "carry" function, f_α, in a full-adder contains an embedded minterm, and that its Boolean differences are covered by those of f_b, where f_b is the function resulting from f_a by eliminating the embedded minterm.

The "carry" function in a full-adder is given by

$$f_a = xy\bar{c} + x\bar{y}c + \bar{x}yc + xyc \qquad (19)$$

(x=augent digit; y = addend digit; c = carry).

It is evident that $m_q^{(3)} = xyc$ is the embedded minterm since it is the only one whose companion set is also in f_a. If xyc is eliminated due to some corruptive mechanism, the resulting function will be

$$f_b = xy\bar{c} + x\bar{y}c + \bar{x}yc \qquad (20)$$

The Boolean differences of f_a and f_b are given by

$$\begin{aligned} D_x f_a &= \bar{y}c + y\bar{c} \\ D_y f_a &= \bar{x}c + x\bar{c} \qquad (21) \\ D_c f_a &= \bar{x}y + x\bar{y} \end{aligned}$$

and

$$\begin{aligned} D_x f_b &= \bar{y}c + y\bar{c} + yc \\ D_y f_b &= \bar{x}c + x\bar{c} + xc \qquad (22) \\ D_c f_b &= \bar{x}y + x\bar{y} + xy \end{aligned}$$

respectively.

From (21) and (22) one can easily see that $D_x f_a \subset D_x f_b$; $D_y f_a \subset D_y f_b$; and $D_c f_a \subset D_c f_b$. This implies that any tests derived from the Boolean differences of f_a will also satisfy f_b. Thus, faults resulting only in the elimination of minterm xyc will go undetected.

It was pointed out that in the case of input-line faults one could employ the Boolean differences of the switching function under test to completely check the operational status of all input lines. The procedure for choosing a minimal test-input sequence capable of exercising the system completely with respect to such faults is outlined in another section.

In the case of internal-node faults, the selection of the test-input sequences will be influenced by the particular hardware implementation used and the use of test-input sequences designed for checking the input lines may not be satisfactory. Figure 1 shows two possible implementations of the "carry" function presented in Example 3. It is easily verified that test-input sequence

$$t_1 = 011 - 001 - 101 - 100 \qquad (23)$$

is sufficient to completely exercise all the input lines. The output (response) of the system due to (23) is normally given by

$$r_1 = 1 - 0 - 1 - 0 \qquad (24)$$

If one of the input lines, say x, is subjected to a (s-a-1) fault, then the response becomes

$$r_1' = 1 - 1 - 1 - 0 \qquad (25)$$

It is evident that a fault at the input lines will destroy the ON-OFF pattern of (24). Similar results are obtained for (s-a-o) faults. It should be pointed out that pattern (25) may result due to internal faults as well, but so far we are interested only in a scheme for detecting faults at the input lines. We turn next to internal faults.

The hardware implementations chosen in Fig. 1 are such that when the nodes are subjected to (s-a-1) faults the fault detection problem becomes trivial. Therefore, only (s-a-o) faults are considered. If we now consider the circuit in Fig.1-a we note that in the case of node N_{a1} the test sequence xyc = 100 — 110 will completely exercise N_{a1} (turns N_{a1} ON and OFF) while all other nodes remain unenergized. Similarly, xyc = 011 — 001 and xyc = 101 — 100 will check nodes N_{a2} and N_{a3}, respectively. Combining the above tests into a single test-input sequence, one has

$$t_a = 011 - 001 - 101 - 100 - 110 \qquad (26)$$

which contains t_1 given by (25). The response normally expected from the circuit in Fig. 1-a due to (26) is

$$r_a = 1 - 0 - 1 - 0 - 1 \qquad (27)$$

When node N_{a1} is subjected to a (s-a-o) fault, output sequence (27) becomes

$$r_a' = 1 - 0 - 1 - 0 - 0 \qquad (28)$$

It is evident from comparing responses (28) and (25) that use of test sequence t_1, given by (23), would not have revealed the presence of this fault in Fig. 1-a. On the other hand, use of test sequence (26) is sufficient for checking both input-line as well as internal faults in Fig. 1-a.

In considering next the other implementation for the "carry" function given in Fig. 1-b, we see that although test sequence (26) is sufficient to check the operational status of nodes N_{b1}, N_{b2} and N_{b3}, a (s-a-o) fault at node N_{b4} will go undetected. In this latter case, the output response is identical to the normally expected output given by (27). This illustrates the point made earlier that the hardware implementation will often influence the choice of test-input sequences used in checking a switching function. One may safely say a detailed analysis of the network under internal-node fault conditions must be made prior to using, for error detection purposes, a test-input sequence designed for checking primarily the input lines. There will be instances in which input-line tests may be sufficient to check completely the entire network for (s-a-o) or (s-a-1) faults, but it does not seem to be an easy way of identifying such cases other than a complete fault analysis of the particular hardware implementation under study.

A complete, minimal test-input sequence is defined to be a sequence of minimal length and capable of exercising completely all the input lines and all the internal nodes of a given switching network. Since the input-line test is included in the overall test, it is evident that the test-input sequence must include at least the code-groups needed in exercising the input lines. Any additional code groups that might be needed must come from a detailed error analysis of the particular network considered. In the following section, an algorithm is presented for deriving complete, minimal test-input sequences for checking the operational integrity of a switching network.

DERIVATION OF COMPLETE TEST-INPUT SEQUENCES

Consider a memoryless discrete parameter system and a node, P, in the system at which one wishes to make observations concerning the system's functional integrity. If F_p is the logic function representing the logical dependency of the state of node P on the input set

$$x = \{x_1, x_2, \ldots, x_n\} \qquad (29)$$

the Boolean differences

D_iF_p, where $i = 1,2,\ldots, n$

denote the conditions under which F_p will depend on the various input variables x_i. Even though we refer to x_i's as input signals, actually they may be, in general, any non-redundant signals within the system which control the logical behavior of node P. If we use

S_{ij} where $i = 1,2,\ldots, n$ and $j = 1,2,\ldots,m$

to represent the code groups (minterm valuations) for which $D_iF_p = 1$, the test-input sequence needed to check the input lines will be constituted of code-group pairs of the form

$$\sigma_{ij} = x_i S_{ij} - \bar{x}_i S_{ij} \qquad (30)$$

where $i = 1,2,\ldots,n$, and S_{ij} are properly selected minterms of the remaining (n-1) variables $(x_1,x_2,\ldots,x_{i-1}, x_{i+1},\ldots, x_n)$ so that the complete set of code groups needed to form the tests stipulated by (30) in exercising the input lines involves a minimum number of codes. In the case of n non-redundant variables, one will need at least n+1 code groups to perform an input-line test. Certain switching functions will need only n+1 code-groups to exercise completely their input lines. This does not, however, imply that such test-input sequences are also sufficient to check the system's internal structure for possible faults. The same applies to test-input sequences of length greater than (n+1). In order to determine whether or not a minimal test-input sequence, designed for checking input lines, is also sufficient to exercise the internal nodes of the system, one must perform a detailed error analysis of the system. It should be pointed out that such an analysis must take into account the fact that there may be more than one minimal test-input sequence capable of exercising completely the input lines.

The following example illustrates the steps involved in obtaining a complete, minimal test-input sequence.

Example 4: Let P be a node in a memoryless switching network whose logical description is given by the function

$$F_p = x_1x_2 + x_1x_3 + x_2x_3$$

Find the set of complete, minimal test-input sequences for checking node P.

Function F_p represents the "carry" of a full-adder and its Boolean differences were given in Example 3 as

$$D_1F_p = \bar{x}_2x_3 + x_2\bar{x}_3$$

$$D_2F_p = \bar{x}_1x_3 + x_1\bar{x}_3$$

and

$$D_3F_p = \bar{x}_1x_2 + x_1\bar{x}_2. \qquad (31)$$

Let

$$S_{11} = \bar{x}_2x_3 \quad , \quad S_{12} = x_2\bar{x}_3$$

$$S_{21} = \bar{x}_1x_3 \quad , \quad S_{22} = x_1\bar{x}_3 \qquad (32)$$

and

$$S_{31} = \bar{x}_1x_2 \quad , \quad S_{32} = x_1\bar{x}_2$$

In view of (32), one may test the input lines by application of code-group pairs as follows:

Line, x_1: $\sigma_{11}=x_1S_{11} - \bar{x}_1S_{11}$ or $\sigma_{12}=x_1S_{12} - \bar{x}_1S_{12}$

Line, x_2: $\sigma_{21}=x_2S_{21} - \bar{x}_2S_{21}$ or $\sigma_{22}=x_2S_{22} - \bar{x}_2S_{22}$ (33)

Line, x_3: $\sigma_{31}=x_3S_{31} - \bar{x}_3S_{31}$ or $\sigma_{32}=x_3S_{32} - \bar{x}_3S_{32}$

By rewriting (33) in terms of x_1, x_2 and x_3, one obtains

Line, x_1: $\sigma_{11}=x_1\bar{x}_2x_3 - \bar{x}_1\bar{x}_2x_3$, $\sigma_{12}=x_1x_2\bar{x}_3 - \bar{x}_1x_2\bar{x}_3$

Line, x_2: $\sigma_{21}=\bar{x}_1x_2x_3 - \bar{x}_1\bar{x}_2x_3$, $\sigma_{22}=x_1x_2\bar{x}_3 - x_1\bar{x}_2\bar{x}_3$

and

Line, x_3: $\sigma_{31}=\bar{x}_1x_2x_3 - \bar{x}_1x_2\bar{x}_3$, $\sigma_{32}=x_1\bar{x}_2x_3 - x_1\bar{x}_2\bar{x}_3$

It is evident from the above listing of code groups that a number of the input-line tests make use of the same groups, and therefore one should attempt to combine tests to form test-input sequence capable of exercising all input lines and utilizing a minimum number of code groups. This is done by inspection, and one such test sequence is

$$t_1 = \bar{x}_1x_2x_3 - \bar{x}_1\bar{x}_2x_3 - x_1\bar{x}_2x_3 - x_1\bar{x}_2\bar{x}_3$$

or in terms of 0 and 1-bits,

$$t_1 = 011 - 001 - 101 - 100$$

This is a test-input sequence of least length (n+1=4). Another possible test-input sequence is

$$t_2 = \bar{x}_1x_2x_3 - \bar{x}_1x_2\bar{x}_3 - x_1x_2\bar{x}_3 - x_1\bar{x}_2\bar{x}_3$$

or $t_2 = 011 - 010 - 110 - 100$

We may now proceed to determine the test-input sequences which will check the network for internal faults using either t_1 or t_2 obtained above as a basis. At this point one must consider the hardware implementation of F_p. An error analysis of Fig. 1-a (where x_1=x,x_2=y,x_3=c) which represents a possible implementation of F_p reveals that when node N_{a1} has failed t_1 will not reveal the presence of such a fault. Similarly, a fault in N_{a3} will not be detected by test-input sequence t_2. All these results are predictable by inspecting t_1 and t_2; one notes that there is no code present in t_1 for exercising node N_{a1} and similarly for t_2 with respect to node N_{a3}. Either one of these sequences may become a complete, minimal test-input sequence by adding either (110-010) to t_1 or (101-100) to t_2. Thus,

$$t_{1c} = 011 - 001 - 101 - 100 - 110 - 010$$

and

$t_{2c} = 011 - 010 - 110 - 100 - 101 - 100$

are two complete, minimal test-input sequences capable of checking the network of Fig. 1-a completely for both input-line and internal faults. Since different faults will result in different output patterns, the response of a system to a test-input sequence may be used as a fault diagnosis tool. It should be pointed out, however, that a certain response may be associated with more than one fault as in the case of Fig. 1-a. If line x_1 is (s-a-o), this fault is equivalent to having nodes N_{a1} and N_{a3} stuck-at-zero. Once such faults are localized the ambiguities are easily resolved by simple tests. Let us next turn to Fig. 1-b which is another hardware implementation of F_p. The presence of the embedded minterm $x_1x_2x_3$ as a separate implementation adds an extra node which must be considered in designing the test-input sequences. From purely network considerations, one sees that code group (111) is the only one which will exercise N_{b4} and it must therefore be included in the test sequence. In this particular case (100 - 111) is an acceptable test for node N_{b4}. Unless there is a specific need for generating such terms, they should be always factored out in so far as their contribution to a function is concerned since this is always possible with embedded minterms.

The algorithm for obtaining complete, minimal-test-input sequences may be stated as follows:

Step 1: Obtain the Boolean differences of the given switching function.

Step 2: Identify all code groups S_{ij} of the remaining (n-1) variables associated with the Boolean differences and form all input-line tests of the form $\sigma_{ij} = x_i S_{ij} - x_i S_{ij}$.

Step 3: Select judiciously the fewest possible number of code groups, x_iS_{ij}, (from those derived in Step 2) which satisfy the input-line tests σ_{ij} for $i = 1,2,\ldots,n$ and for at least one value of j. If the choice of code groups is such that

$$\sigma_{1j_1} = (g_1 - g_2), \sigma_{2j_2} = (g_2 - g_3), \ldots, \sigma_{nj_n} = (g_n - g_{n+1})$$

then the resulting test-input sequence may be written as

$$t = g_1 - g_2 - \cdots - g_{n+1}$$

Such a sequence is a least-length test-input sequence and may or may not be always possible. When least-length test-input sequences are not feasible, one may still derive a minimal-length test sequence by judicious choice of code groups. The arrangement of the code groups in the final test-input sequence must be such that the output response becomes either 101010 --- or 010101---, which are easily detectable forms. The test-sequences obtained in this Step are capable of exercising completely all input lines.

Step 4: Once a minimal test-input sequence capable of checking all input lines has been obtained, the next step involves a complete error analysis of the actual hardware implementation of the function under consideration. This is done by first establishing the dependence of each node on the input signals through the use of Boolean differences. These results will yield the information necessary for exercising the various nodes. Once the code groups needed to exercise each and every node are obtained, one must proceed to incorporate these codes in the test-input sequence obtained in Step 3.

Step 5: In order to ensure that the designed test-input sequence in Step 4 is completely exercising the network's input lines and internal nodes, set up a "truth table" involving the nodes as variables which may be assigned the two "fault values" (s-a-o) and (s-a-1). One may then apply the designed test-input sequence and observe the output response for each of the possible 3^N-1 fault conditions, where N is the number of nodes. In most practical systems the number of nodes N is much less than the number of input code groups 2^n, where n is the number of input variables. Because of this, fault diagnosis based on such a "truth table" scheme is quite promising.

Although the procedure of obtaining complete, minimal test-input sequences was based on switching functions describing memoryless systems, the approach applies to systems endowed with memory properties provided that the initial state of the system is known.

CONCLUSIONS

A simple scheme for testing logic hardware has been presented, and complete minimal test-input sequences for exercising all input lines and internal nodes of a switching network was shown to be always feasible by making use of Boolean differences. These test sequences may be stored and subsequently called upon to check periodically a given system for possible input-line and internal-node faults. The faults may be either single or multiple. Some of the advantages of this error-checking scheme are: i/ the simplicity of its implementation, ii/ its suitability for use with any logical hardware regardless of complexity; and iii/ its potential for fault localization since it exercises selected transmission paths in a well-prescribed manner.

REFERENCES

1. Hamming, R. W., "Error Detecting and Error Correcting Codes," Bell System Tech. J., Vol. 29, pp. 147-160 (1950).

2. Peterson, W. W., and M. O. Rubin, "On Codes for Checking Logical Operations," IBM J., Vol. 3, pp. 163-168, (April 1959).

3. Pierce, W. H., R. H. Wilcox, and W. C. Mann, "Redundancy Techniques for Computing Systems," Spartan Book, Inc., Washington, D.C., (1962).

4. Winograd, S., "Redundancy and Complexity of Logical Elements," Information Control, Vol. 5, pp. 177-194, (1963).

5. Sellers, F. F., M. Y. Hsiao, and L. W. Bearnson, "Error Detecting Logic for Digital Computers," McGraw-Hill Book Company, New York, 1968.

6. Marinos, P. N., "Boolean Differences as a Means for Deriving Complete Sets of Input Test Sequences for Fault Diagnosis in Switching Systems," Bell Tel. Labs, Tech. Memo. No. MM-69-6313-1, (August 1969).

7. Sellers, F. F , M. Y. Hsiao, and L. W. Bearnson, "Analyzing Errors with The Boolean Difference," IEEE Trans. on Computers, Vol. C-17, No. 7, pp. 676-683, July (1968).

8. Yen, T. Y., "A Method of Automatic Fault-detection test generation for Four-phase MOS LSI Circuits," Proc. SJCC, pp. 215-220, 1969.

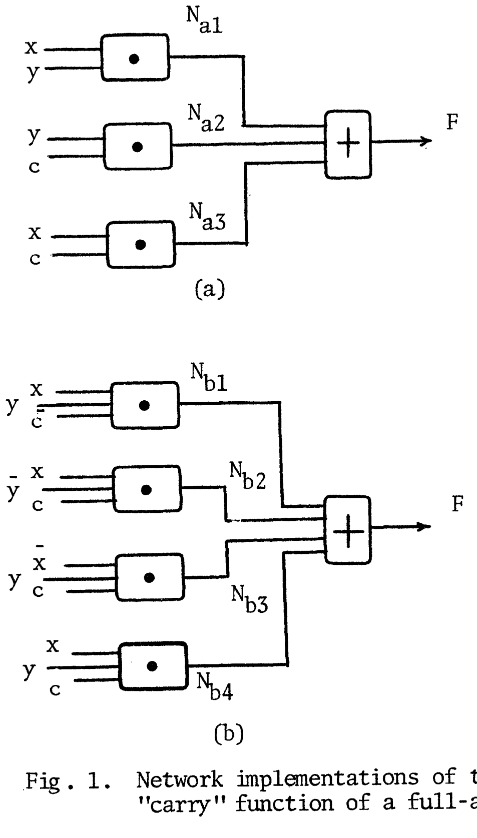


Fig. 1. Network implementations of the "carry" function of a full-adder.

AN ALGORITHM TO COMPUTE A TEST TO DISTINGUISH BETWEEN TWO FAILURES IN A LOGIC CIRCUIT

J. Paul Roth, IBM Thomas J. Watson Research Center, Yorktown Heights, New York

ABSTRACT:

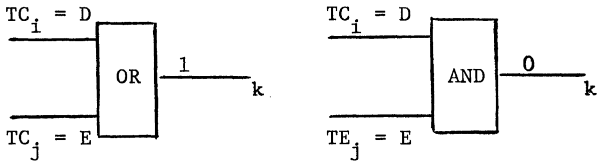
Algorithms are known [1] to compute a test to detect a failure in logic circuits. In practice however a single test or set of tests will detect possibly many failures and it is necessary, for purposes of repair, to identify just which failure has occurred (at least up to indistinguishable equivalence classes). This note extends the D-algorithm to accomplish this purpose. It applies to combinational and by [2], to sequential circuits.

It is assumed that the reader is familiar with the D-algorithm [1] which computes a test, if one exists, to detect any given failure in a logic circuit L. By [2] it is no loss of generality to assume that L is acyclic (hence is combinational). As usual we assume that the category of failures we are dealing with consist of the single stuck-at-1, stuck-at-0 variety.

Let α and β denote two failures in L and suppose that it is desired to compute a test T to distinguish between them. Clearly a necessary condition on T is that either one of them, say α, must yield a response under T that is different from that of the perfect circuit. Accordingly we use the D-algorithm, appropriately modified, to compute a test that will distinguish the logic circuit with failure α from the perfect logic circuit and at the same time distinguish it from β. If this is not possible, we reverse the roles of α and β. If neither is possible then it is shown that α and β are indistinguishable failures. The procedure checks, at each stage of the extension of the developing test cube TC, for failure α, whether or not the distinguishing signals D and E from α and β respectively are themselves mutually distinguishable. Thus in general it is not necessary to go through the algorithm exhaustively to ascertain that no distinguishing test exists: it will ordinarily be so determined *in media res*.

The Interactive Method

We assume the failures α and β are on distinct lines. Thus the initial D-chain TC and E-chain TE are disjoint. Thus in extending these chains we need to come to terms with the case when line k has for its input lines i and j, $TC_i = D$ or $\bar{D}$ and $TE_j = E$ or $\bar{E}$. We shall discuss the case of 2-input AND- and OR-blocks. See figure below.



$$TC_i = x$$

$$TE_j = x$$

Since the occurrence of failures α and β are mutually exclusive, when $D = 0$ (α has failed) then $E = 1$ (β has not failed) and vice versa. Thus if the block is an OR, the input is identically 1; if an AND, identically 0. Hence $TC_k = TE_k = 0$ or 1 depending upon whether the block is an AND or an OR. Clearly the same result holds when $TC_i = \bar{D}$ and $TE_j = \bar{E}$.

Now consider the case when $TC_i = D$ and $TE_j = \bar{E}$. For this purpose suppose that neither failures α and β have occurred. Then $TC_i = D = 1$ and $TE_j = \bar{E} = 0$ so that output on line k is a 0 for the block an AND. Let us compare this with the case when failure α has occurred but β has not. Then $D = 0$ and $\bar{E} = 0$, so that the output for the AND-block will also be zero, so that TC_k should be 0. For the case α has not failed but β has, output k is 1 so that $TE_k = \bar{E}$. Hence for this case the D-chain TC does not propagate through to line k, but the E-chain TE does. In similar fashion if the block were an OR, TC_k would be D and $TE_k = 1$. It is easy to extend this analysis to arbitrary logic blocks.

Lemma 1. In logic circuit L subject to failure α or failure β but not both, suppose that, corresponding to any input pattern T, the value on some line ℓ is fixed, i.e., $TC(\ell) = i = 1$ or 0. Then it is not possible that $TE(\ell) = E$ or $\bar{E}$. Specifically it is necessary that $TE(\ell) = i$.

Proof. For $TC(\ell)$ to have the fixed value i means that it assumes this value independently of whether failure α occurs or not, i.e. D = 1 or 0. But since failures α and β are mutually exclusive and mutually exhaustive, the value E (or $\bar{E}$) clearly changes dependent upon this circumstance. Hence the signal on line ℓ cannot be constant. Q.E.D.

Note that in lemma 1 it is not necessary that the input pattern T assumed be a primary input pattern. Thus these criteria will be used in the D-algorithm so modified at each stage of extension of the D-chain: if a line has inputs D or $\bar{D}$ and E or $\bar{E}$, then its own signal will be determined according to the above table and reckoning and, in any event, it will be removed from the D-frontier of the developing TC.

If on the other hand $TC(\ell) = 1$ or 0 and $TE(\ell) = E$ or $\bar{E}$, this is a contradiction and the last decision prior to this circumstance must accordingly be undone or if there remain no subsequent alternate choices, then the quest for a distinguishing test will be terminated as unsuccessful.

Description of Distinguishing Algorithm.

First it will be attempted to compute a test distinguishing between α and β for which the output pattern for failure α is different from that of the perfect circuit. If this is not possible then a distinguishing test is sought for which the failure pattern of β is different from that of the perfect circuit. If neither is possible then clearly no test exists distinguishing between α and β .

The algorithm then begins with the imposition of a D on the line manifesting failure α :

$$TC(0) = \overset{\alpha}{D}$$

(in contradistinction with the method of [1] and [3] we do not start out with a pdcf-unless there is only one pdcf-but instead merely impose D on the appropriate line and then at a later time, in CONSISTENCY, select an appropriate pdcf, as in [2]: this procedure will in general be more efficient). At each stage of development of the test cube TC corresponding to failure α we compute the corresponding implied development of the test cube TE for failure β .

Having developed the test cubes for failures α and β up to the nth step, TC(n) and TE(n), we select for the line ℓ appropriately chosen from the D-frontier of TC(n) a primitive D-cube pdc and D-intersect it with TC(n), to form

$$TC^*(n+1) = TC(n) \cap pdc .$$

If this intersection is empty we proceed as in the D-algorithm to another choice of pdc replacing the present one, or to another D-frontier line ℓ if no other pdc's are available. If this intersection is nonempty then all possible implications of this change are made, to form TC(n+1). Then all possible corresponding implications of this change are made on TE(n), to form TE(n+1) . Then any possible conflicts are ascertained between (the new parts of) TC(n+1) and TE(n+1) as in the above lemma. If any conflicts occur the development is restored to the last decision point, (Lemma 1). If D-($\bar{D}$-) and E-($\bar{E}$-) occurs on the inputs to a given line k, TC_k and TE_k are computed by the interactive method. Then the procedure continues inductively until the D-chain from α reaches a primary output.

Thence one goes through the CONSISTENCY operation to complete the development of a test cube TC for failure α ; at each stage the implied development of the test cube TE for failure β is made and inconsistencies between these two is checked as above. It is clear that for any primary input coordinate of the final TC that is x, any choice may be made without a conflict arising between TC and TE.

If no test is found of the type that the response for failure α is different from that of the perfect circuit, then one reverses the roles of α and β and seeks a distinguishing test such that the response of failure β differs from that of the perfect circuit. If no such test exists either, then there is no test distinguishing α from β.

This completes the description of the distinguishing algorithm. We must now prove the effectiveness of the algorithm.

Theorem. Given failures α and β in a logic circuit L, the distinguishing algorithm will compute a test distinguishing them if such a test exists.

Proof. The proof is very similar to that for the D-algorithm and will be modelled after it. Let T be a distinguishing test and $c(T,\alpha)$ and $c(T,\beta)$ be the D-cubes of test T and failures α and β respectively; as above, however, let E and $\bar{E}$ be substituted for D and $\bar{D}$ respectively in $c(T,\beta)$. Assume, with no loss of generality, that the response of α to test T is different from that of the perfect circuit and thus that there is a connected D-chain from α to some primary output which is not also reached by the E-chain emanating from β . The distinguishing algorithm, as with the D-algorithm, establishes a lexicographical ordering of D-chains, the order of formation by the algorithm itself; let it be assumed that the algorithm forms no D-chain defining a distinguishing test $T' \neq T$ before constructing $c(T,\alpha)$ and $c(T,\beta)$, for otherwise the theorem is already proved.

We start out with a D on the line corresponding to failure α , $TC(0) = \overset{\alpha}{D}$. The algorithm selects a line on the D-frontier of TC(0) and selects a pdc p therefor; if this line and pdc does not correspond to that of $c(T,\alpha)$, this means that this choice would run into a "blind alley" for by hypothesis there is no $c(T',\alpha)$, $T' \neq T$, constructed by the algorithm before c(T,α). Therefore we choose the pdc p defined by c(T,α) and form the D-intersection

$$TC^*(1) = TC(0) \cap p$$

Since both TC(0) and p D-contain c(T,α), so does their D-intersection and this intersection is thus nonempty, by Lemma 5 of [3]. Likewise the D-intersections TC(1) ⊃ TC*(1) implied by TC*(1) must also D-contain c(T,α) for the same implications of TC(1) would also be made by c(T,α). Likewise we form the E-chain TE(1), implied by TC(1), emanating from the failure β. Clearly TE(1) ⊃ c(T,β) since TE(1) is implied by only TC*(1) which is only a part of c(T,α).

Suppose then that TC(n) ⊃ (T,α) and TE(n) ⊃ c(T,β) have been formed. Let q be the primitive D-cube or primitive singular cube next selected by the algorithm; it may be assumed that q contains c(T,α) since by hypothesis all test cubes constructed in lexicographical order before c(T,α) fail. Thus their D-intersection

$$TC^*(n+1) = TC(n) \cap q$$

also D-contains c(T,α) as well as TC(n+1), the result of all implications from TC*(n+1). Likewise we form TE(n+1) as a result of TE(n) and the new implications from TC(n+1); as a result of this possibly, by Lemma 1 and the interactive method, certain members of the D-frontier will be eliminated. Thus the distinguishing algorithm proceeds inductively until the test cube TC for failure α is justified in terms of primary input patterns. Furthermore this pattern T*, being in general a cube (i.e. having some x-coordinates) will contain T and will by construction distinguish α from β. Q.E.D.

Efficiency of Algorithm.

It is estimated that the time to compute a test distinguishing between two failures will be somewhat longer than although the same order of magnitude as the D-algorithm, to compute a test to detect any given failure, for there are only a few extra operations in the distinguishing algorithm over the D-algorithm in the basic operation of extending the D-chain by D-intersection. Thus I would estimate the distinguishing algorithm to be slower than the D-algorithm at most by a factor of 2.

REFERENCES:

1. Roth, J. P., Bouricius, W. G., and Schneider, P. R., "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits," IEEE Trans. on Electronic Computers, EC-16 (Oct. 1967).

2. Duke, K. A., Putzolu, G. R., and Roth, J. P., "An Algorithm and a Program for Generation of Test Patterns for Sequential Circuits," IBM Research Report (RC 2639) to appear.

3. Roth, J. P., "Diagnosis of Automata Failures: A Calculus and a Method," IBM J. of Research and Development, 10, 278-291 (July 1966).

4. Hornbuckle, G. D., and Spann, R. N., "Diagnosis of Single-Gate Failures in Combinational Circuits," IEEE Trans. on Electronic Computers, C-18, 216-220 (March 1969).

A PATH SENSITIZING ALGORITHM FOR DIAGNOSIS OF BINARY SEQUENTIAL LOGIC

Thomas A. Kriz
Automation Systems Development Department
International Business Machines Corporation
Federal Systems Division
Owego, New York 13827

ABSTRACT

The "path sensitizing" method which generates "distinguishing sequences" for a set of transmission path submachines in a logic model to accomplish diagnosis is extended to the case of the binary sequential machine. The extension, which is discussed in terms of D-chain test theory, involves execution of an appropriate set of sequentially chained D-cubes to sensitize and test a path in a sequential machine. Successful implementation of the technique requires an algorithmic capability to pass part of the structure along a "path" into a particular known state. A portion of the algorithm discussed here accomplishes this with a D-type calculus for state tracing in a binary equivalent state transition model of the sequential logic structure. Another portion of the algorithm uses the same calculus to determine the sequence of controllable input states which are needed to enable or inhibit propagation of a state change along the "path".

INTRODUCTION

Historically, the older and more prominent techniques for diagnosis of a sequential machine have involved the generation of a "diagnosing" or "distinguishing" sequence to partition a collection of complete machines into equivalence classes. This approach, initially developed by Moore[1], has been exploited for analysis of combinational and sequential logic by a number of authors[2-6]. A more recently developed alternative approach is that of "path sensitizing" which, thus far, has been described only for the case of combinational logic[7-9]. This technique essentially concerns generation of "distinguishing sequences" for a set of sensitized transmission path submachines in a logic model to accomplish diagnosis. The machine identification concepts of the "path sensitizing" method have been discussed in an earlier paper by the author[10]. The purpose of this paper is to present an algorithm which extends the "path sensitizing" diagnosis concept to the case of binary sequential logic.

Conceptually, the extension of the "path sensitizing" diagnosis technique from the combinational logic area to the sequential logic area is a relatively easy matter. For the sequential logic structure, however, since the operation of the structure is dependent upon memory variable states as well as non-memory variables, one must also consider the state transition characteristics of the memory variables in order to apply the technique. Thus, one must consider embryonic tests for the memory blocks as well as those of the gating blocks in the structure. As in a combinational net, it must also be assumed for successful application that every part of the logic structure is controllable and, in some sense, observable. Within this framework, then, a brief recipe for "path sensitizing" might be stated as follows. First of all, an appropriate set of propagation paths in the structure must be chosen. Next, the input sequences which will appropriately initialize the state of the sequential structure attendant to and along the paths must be determined. Finally, the input sequences which exercise the transmission path submachines by enabling or inhibiting propagation of a binary state change along the paths must be found.

In applying these sets of input signal sequences to test a model of the logic structure, the following fault isolation philosophy can be applied. If an object under test gives the correct response, the constituents of the path may be considered to be "good". This, of course, presumes a very small likelihood of correct response when a failure exists. If a prescribed state change does not propagate or is not inhibited as required, a stuck-at-one or a stuck-at-zero failure may be presumed to exist in a logic element (or elements) along that path or along paths which were used to initialize and sensitize that path. By use of tests for intersecting paths, then, (if they exist), one can attempt to resolve the location of the failure to a lower level.

Due to the existence of memory in the sequential model, one is faced with the problem of initializing the state of the memory elements and forcing the propagation of state changes through them, as well as the combinational portions of logic. Resultantly, one must devise an algorithm which considers the state transition properties of the sequential logic in terms of the flip-flop state control conditions. Such an algorithm is given in the following section. This algorithm is essentially a D-type-algorithm which includes additional complexity in order to model binary memory variables in a sequential machine.

ALGORITHM

The "path sensitizing" algorithm given in this section assumes the following about the nature of a sequential logic structure:

1. Memory elements are binary latches such as J-K/R-S complementing type flip-flops with unclocked inputs set, S, and reset, R, and clocked inputs for set, J, and reset, K, which are gated during a true-to-false transition of a clock input C. A sequential logic structure composed of these memory elements and combinational gates such as AND, OR, NAND etc, is assumed. It is further assumed that all binary memory elements in the structure are explicitly identified and will be treated as such.

2. The state-transition characteristics do not depend upon critical races in the combinational logic portions of the structure which attach to the memory element set. It is assumed in the case of synchronous state transitions that the present state of a memory variable is dependent only upon previous state of the sequential machine, and present controllable input states.

3. The state transition characteristics of a diagnosable path are such that the memory elements along the particular propagation path may be initialized, and state changes propagated along that path using a controllable input signal sequence which does not permit state changes to simultaneously enter the path via other routes. The significance of this requirement will be fully discussed later.

4. The logic structure is such that it can be diagnosed with a set of preset diagnosing experiments[11]. Thus, logic structures which require adaptive diagnosing experiments are precluded.

An outline of the sequential logic path sensitizing algorithm is given in Fig. 1. Procedural details of the algorithm are presented in the following discussion. Several illustrative examples which demonstrate the use of the algorithm will be discussed in the next section.

As an initial step in explaining the algorithm shown in Fig. 1, some consideration must be given to the question of what constitutes an embryonic test for the memory element set which was specified. The nature of these tests will be described in terms of primitive D-cubes. Embryonic tests for combinational logic block primitive D-cubes, and the D-cube terminology which will be used have previously been discussed by Roth[8]. This discussion is necessary because the first several blocks of the algorithm outline given in Fig. 1 make use of all such primitive D-cube information to define D-chains corresponding to propagation paths and sensitizing conditions in a sequential logic structure.

The discussion begins with a consideration of the state control conditions of the flip-flop. First of all, for the J-K/R-S type flip-flop with true output F and inputs J, K, R, S, C, it should be noted that the SET state control condition in Boolean equation form may be stated as

$$\sigma = \mathrm{S\bar{R}}\ \overline{(\mathrm{J\bar{K}C/\bar{C}})} + \mathrm{J\bar{K}\bar{S}\bar{R}C/\bar{C}}, \tag{1}$$

the RESET state control condition as

$$\rho = \mathrm{\bar{S}R}\ \overline{(\mathrm{\bar{J}KC/\bar{C}})} + \mathrm{\bar{J}K\bar{S}\bar{R}C/\bar{C}}, \tag{2}$$

the TOGGLE state control condition as

$$\tau = \mathrm{JK\bar{S}\bar{R}C/\bar{C}} \tag{3}$$

and the AT REST state control condition as

$$\alpha = \mathrm{\bar{J}\bar{K}\bar{S}\bar{R}} + \mathrm{\bar{S}\bar{R}}\ \overline{(\mathrm{C/\bar{C}})} \tag{4}$$

where $\bar{\mathrm{X}}$ is the logical inverse of X, and $\mathrm{A/\bar{A}}$ is the true-to-false transition of A. A complete definition of the useful state transition characteristics of the flip-flop from time i-1 to time i may now be stated in terms of (1) - (4). These are Boolean state transition equations

$$\mathrm{F(i)/1} = \sigma + \tau\ \bar{\mathrm{F}}(\mathrm{i}-1) + \alpha\ \mathrm{F}(\mathrm{i}-1) \tag{5}$$

and

$$\mathrm{F(i)/0} = \rho + \tau\ \mathrm{F}(\mathrm{i}-1) + \alpha\ \bar{\mathrm{F}}(\mathrm{i}-1) \tag{6}$$

where F(i)/1 indicates F is true at time i, and F(i)/0 indicates F is false at time i. The primitive D-cubes which describe an embryonic test for the flip-flop can be defined using (1) - (6). These assume that an attempt has been made to initially drive the flip-flop to a particular initial state. A list of these D-cubes for the flip-flop and the terms of (1) - (4) that they correspond to is given in Table I. In these cubes, 1, 0, X, 1/0, 0/1 and D, $\bar{\mathrm{D}}$ are respectively true, false, don't care, true-to-false transition, false-to-true transition and D-states of a variable. As can be seen in the table, the D-cubes which trigger a state change are associated with the σ, ρ, and τ state control conditions and those which inhibit state change are associated with α. Embryonic tests which correspond to the primitive D-cubes in Table I may be subsequently interpreted in a manner somewhat similar to that used to interpret combinational D-cubes. For example, the first D-cube listed in Table I, D D 0 X X 0 indicates that with D = 1 for S and F = 0 initially, the cube constitutes an embryonic test for F or S stuck at 0, and alternatively, with $\bar{\mathrm{D}}$ = 0 for S and F = 0 initially in $\bar{\mathrm{D}}\,\bar{\mathrm{D}}$ 0 X X 0, an embryonic test for F or S stuck at one is defined. It may also be noted that the cube D D 0 X X 0 implies the cube $\bar{\mathrm{D}}\,\bar{\mathrm{D}}$ 0 X X 0 since the second cube is simply the first one with its D-states inverted. Thus, the inhibit D-cubes do not have to be explicitly represented if one has the trigger D-cube set. The significance of this fact will be more fully discussed later.

If all the D-cubes in Table I are interpreted as the one mentioned above, one finds that there are more than enough D-cubes to completely test every variable stuck at 1 or 0. These primitive D-cubes, in a manner similar to the combinational logic case, may be used to define D-chains for more complicated structures which contain many memory elements. In doing this, primitive D-cubes of different memory elements may be chained to each other directly or via D-chains through combinational logic sections to define propagation paths in a sequential logic structure which may be used for testing. Indeed, this is exactly what occurs during execution of blocks I and II of the algorithm defined in Fig. 1. The particular details of these blocks will now be discussed.

As can be seen in Fig. 1, the first step of the sequential path sensitizing algorithm concerns finding the combinational logic subsets which drive flip-flops or which provide sequential machine outputs. This information is obtained by searching logic model data which describes each logic block and how it is connected to other logic blocks. The flip-flop and the combinational logic blocks which drive into it are, then, taken together to form a large binary memory block which will be termed a super flip-flop.Likewise, a combinational logic subset which provides a machine output is consolidated to form a super-output block. In forming these super blocks it should be noted that a particular primitive block may simultaneously be contained in more than one super block. This causes no difficulty in the process of generating tests which will be described.

After the super blocks which make up the sequential machine have been defined, one of the combinational path sensitizing methods [7-9] is used to help define D-cube or propagation path sensitizing conditions for each super block. In the case of the super output block, the use of these methods is straightforward. For the super flip-flops, however, one must chain the primitive D-cubes of an elementary flip-flop to the D-chains in the combinational logic subset which drives it. This requires adherence to a procedure which is as follows. For each primitive memory element D-cube, first, the D-chains of the combinational logic subset which generates the D-term variable must be found using one of the combinational logic path sensitizing methods. [7-9] The D-states of this cube are appropriately modified to be consistent with the D-input state of the memory block. Finally, the 1-0-X cubes for the combinational logic subsets which generate the sensitizing terms in the memory block D-cube are found and added to it.

As shown in block II of Fig. 1, the next step involves definition of the D-chains locally on transmission paths in the sequential machine itself. This is accomplished by tracing and chaining together the D-cubes of the super logic blocks which were processed in block I. The chaining process is similar to that which may be used for chaining of combinational logic cubes. [8] The tracing process begins by considering a super logic block D-cube which generates an observable machine output. If there is an input coordinate D-state in this cube generated by a super flip-flop, the D-cubes of that super flip-flop having an output D-state matched to the input D-state of the first D-cube are chained to it. This process is then continued for each of the new D-cubes, and so on, until one traces to a D-cube whose input coordinate D-state is a controllable sequential machine input or a memory element which is already in the D-chain which is being constructed. The latter case occurs when one encounters a logical loop in the sequential machine. During the trace process, ordered lists of super logic block D-cubes which correspond to a sequential machine D-chain are constructed. The peculiarities of this kind of D-chain will be discussed shortly. The procedure just described may be applied to all observable machine output super block cubes in order to generate every D-chain in the sequential machine structure. Obviously, only a covering set of these need be utilized to detect faults in a sequential machine.

It is evident from the foregoing that the sequential logic D-chains which are constructed with super logic block D-cubes are somewhat different than the D-chains which have been described for combinational logic nets [8]. The difference between them and the reasons for it will now be considered. In the case of a combinational net, since there is no internal memory, a response to an input stimulus must occur as immediately as permitted by the propagation delay characteristics of the net. Consequently, one can construct a single D-cube for a chain of primitive cubes which completely defines the propagation path which will be used for a test and the state of every variable which is necessary to sensitize that path. For the sequential machine, on the other hand, since it generally possesses an unobservable internal memory (except in degenerate cases where all memory states are observable), the response to an input stimulus which corresponds to the content of a single D-cube can "get stuck" internally and be unobservable. In a similar manner, the input stimulus itself may "get stuck" internally. Thus, additional D-cubes which allow continued propagation must be executed in order to allow observation of a response. These extra D-cubes might well be defined in terms of conditions which are local to each binary memory element along the propagation path. Indeed, such is the procedure which was used in the preceding paragraph to define sequential machine D-chains. This kind of D-chain is a set of D-cubes, executed sequentially, which do completely define a propagation path, but do not define sensitizing conditions except in a manner which is local to each memory element. Therefore, some additional means is needed to define the sensitizing conditions from a controllable machine input viewpoint so as to have an initializing and exercising sequence of controllable input signal states which allow one to test the logic blocks on a transmission path. Such a means is provided by the binary equivalent state transition model which will be discussed in the following. This model uses a table of local super flip-flop state transition conditions and a state tracing calculus to generate initializing and exercising sequences for the sequential D-chains which are generated during block II of the sequential path sensitizing algorithm.

One part of the equivalent binary state transition model is a table of local state transition conditions for all super logic blocks such as shown in Table III. This table corresponds to an example structure which will be discussed in the next section. As shown in Table III, the state transition information is a covering set of cubes for the σ, ρ, τ and α state control equations in (1) - (4) for each super block. The expressions for σ, ρ and τ state control conditions are listed in a semi-D-cube form so that the table may also be used for constructing sequential D-chains as described previously. The trigger D-cubes are explicitly defined and inhibit D-cubes are implied by the format which is given. It may be seen in Table III that the coordinates within each cube are grouped in terms of output variables, internal variables, and controllable input variables. This is done to facilitate initiation and termination of tracing processes in the table which will be illustrated later. The set of cubes for a particular output D-state and the cubes of its α condition are also arranged in a hierarchical order which corresponds inversely to the number of non-X state memory variable coordinate terms in the cube. This facilitates convenient generation of initializing sequences as will also be illustrated later.

It may also be seen in Table III that some state control condition cubes contain d, $\bar{d}$, T, $\bar{T}$, t, $\bar{t}$, A, and $\bar{A}$ coordinate terms which are used to identify the peculiar nature of an output or input variable under the influence of that particular state control condition cube. For example, d is used to identify an input D coordinate which allows a memory element to be set or reset asynchronously; $\bar{d}$, alternatively, identifies an asynchronous $\bar{D}$ coordinate. T, t, A and the logical inverse of them, on the other hand, are used to designate the special character of memory output variable coordinates in a cube. T corresponds to a toggle mode 1/0 transition of an output variable and $\bar{T}$ to a 0/1 transition. The symbol A is used to identify the output variable of a cube which causes the present state of a memory element to remain the same as its previous state. In this case A defines an "at rest" condition in the "1" state and $\bar{A}$ defines a "0" at rest state. The symbol t identifies an output variable which may change or remain at rest depending upon what the previous state of the memory element was. In this case, t corresponds to a memory variable with a "0" final state and $\bar{t}$ to that with a "1" final state. The utility of the special symbols will now be discussed in context with the state trace calculus.

A second ingredient in the equivalent binary state transition model is a state trace calculus which allows one to determine machine level state transition information from the local state transition information given in the table just described. This calculus makes use of an operator technique called the \$ operator. The \$ operator operates upon symbols of the same coordinate in a set of cubes taken from the super logic block state transition table. The operator is both commutative and associative. It generates, for pairwise operations, the results as shown in Table II. In executing a \$ operation, a * type state may be generated which corresponds to a prohibited set of sequential machine state conditions. This indicates that the set of cubes which were operated upon are not consistent with each other. A different set of cubes must then be selected. Other states which occur are those as previously defined depending upon which combination of symbols are taken. The super logic block state transition table and the \$ calculus constitute the equivalent state transition model identified in block III of Fig. 1. The use of this model to obtain controllable input, path sensitizing test information for the sequential machine D-chains will now be described.

For the kind of sequential D-chain diagnosis which is being described, the fundamental plan of attack involves enabling or inhibiting the propagation of a state change along a path to effect diagnosis of the logic block variables on that path. It is required that the appropriate memory variables of the machine be initialized and subsequently exercised to make the propagation of this state change observable. For each sequential D-chain, two tests must be conducted. In one test, (Type A) the controllable input D-coordinate term of the last cube in the D-chain is set to a state which potentially allows the state change to propagate. This test is used to determine if any of the set of D-variables in the chain are stuck in a state which inhibits propagation of a state change. Alternatively, for the second test, (Type B) the controllable input D-coordinate term of the last cube in the D-chain is set to the other state which inhibits propagation of the state change if no fault exists. This test is used to determine if any of the set of D-variables in the chain are stuck in a state which causes a propagated state change when there should be none in the good copy. In order to execute these tests, the initializing and exercising sequences which were mentioned must be determined for the sequential D-chains which are utilized. The generation of the exercising sequences for the sequential D-chains will be considered first. In determining the sequence of controllable input signal states and machine states which constitute an exercising sequence for the path, the basic problem is, given that the machine is in one state, what state did it come from, and what set of controllable input variable states permitted the transition to occur? Thus, the problem is a kind of backward "homing experiment" problem. In using the \$ calculus to determine this information about the exercising sequence of a particular sequential D-chain, one begins by considering the final state of the process which is described, at least in part, by the first D-cube in the chain starting with the machine output end of the D-chain. It is assumed here that the D-cubes have a format like that used in the table of local state transition conditions. If this first D-cube has a d-type input term, asynchronous propagation from the next D-cube is indicated. Thus, a \$ operation between these two cubes must be executed to yield a single cube which defines the final state of the exercising sequence unless there is also another asynchronous propagation into this set. This, then, indicates the need of a special procedure for sets of D-cubes in a D-chain which are asynchronously chained. The member D-cubes of these asynchronous sets must be combined using the $ operator to yield a single D-cube for each group. These cubes plus the other D-cubes with synchronous input may then be considered using the $ calculus to determine the exercising sequence. The non-X terms of the first cube (starting with the machine output end of the D-chain) indentify the nature of the previous machine state and controllable input signal states as well as the previous D-cube in the chain. The previous binary state interpretation used for particular symbols in the cube during exercise sequence generation are as follows: d, D, T, A, and 1 indicate a true state for a variable; $\bar{d}$, $\bar{D}$, $\bar{T}$, $\bar{A}$, and 0 indicate a false state; D/0, 1/0 indicate a true to false transition; $\bar{D}$/1, 0/1 indicate a false to true transition; and X indicates a don't care state. The symbols t and $\bar{t}$ are interpreted as requiring a previous "don't care" state except in the case of an output variable of a D-cube which is presently being executed where t indicates a previous true state and $\bar{t}$ indicates a previous false state. Cubes which yield these states and the previous D-cube are then taken from the local state transition list and combined using the $ operator. If an * results from this operation, a different combination of cubes must be used. If no * occurs, the resulting cube which is generated by the use of the $ operator on this group of cubes will then define the nature of the next previous machine state and the input signal states which caused the state transition. The procedure just described is subsequently continued until finally there are no unprocessed D-cubes left in the sequential D-chain at which time the backward state trace process for the D-chain exercising sequence is ended. During the process, a sequence of controllable input signal states which exercises the path defined by the sequential D-chain is generated. The exact details of this process which cannot be adequately defined in a verbal fashion will be illustrated in the next section by several examples.

The use of the $ calculus to determine the sequence of controllable input variable states which will initialize the state of the path (block IV, Fig. 1) is somewhat similar to that used to determine the exercising sequence. In this case, however, a set of states as required by the initial state of an exercise sequence for a D-chain is used as the final machine state to begin a backward state trace process. This backward state trace is continued until a complete set of memory variable don't care conditions using the $ calculus is reached. As the backward state trace process progresses, the input sequence for accomplishing initialization of the sequential D-chain is generated. The previous state interpretation during this process is similar to that used for exercise generation except that t and $\bar{t}$ always indicate a don't care condition. The details of this process will also be illustrated by several examples given in the next section.

Aside from the description of the sequential path sensitizing algorithm, some additional mention should be made of the restrictions which were given for the algorithm at the beginning of this section. One of these concerns paths which permit a state change to enter via different routes for the same controllable input signal sequence. As will be demonstrated by the example in the next section, this kind of path (or D-chain) can exist for practical sequential logic structures. It will also be seen, however, that if the path cannot be diagnosed as it is, it can be made diagnosable by the addition of an observable test point.

The foregoing discussion was concentrated upon J-K/R-S type flip-flops. This memory element set was chosen primarily because it contains both clocked and unclock inputs which are typical of most switching circuit memory elements. Quite evidently, other memory element circuits may also be included if they can be modeled with D-cubes without violating other restrictions which were given.

A final comment concerns sequential logical structures which require an adaptive initializing or exercising sequence. Typically, this might be a counter type structure with a cyclic state transition characteristic. Though this kind of structure cannot be handled directly with the algorithm given here, there is a simple hardware change which can make the structure diagnosable. This involves the addition of hardware which will allow one to set the state of the memory elements which have the cyclic transition characteristic to at least one particular state with a preset sequence. Subsequently, the algorithm given here could be used to diagnose the structure.

ILLUSTRATIVE EXAMPLES

Several simple sequential machines which will be used to illustrate the sequential path sensitizing algorithm discussed in the previous section are shown in Figs. 2 and 3. The machine in Fig. 2, called Example A, is simply a one bit shift register with shift clock input from a one-bit counter. The Example B machine in Fig. 3, on the other hand, is a two-bit circulating register with gated input and output. A superfluous flip-flop which provides an unclocked set input to the register has been added to this circuit so as to demonstrate the use of the $ calculus for asynchronously coupled logic. Initially, the super flip-flop state control conditions for these sequential machines will be identified. Subsequently, some of the paths or D-chains for each machine will be generated and converted to initializing and exercising test sequences using the $ calculus.

Example A shown in Fig. 2 is considered first. As an initial step, one must find the combinational logic which drives into each flip-flop and join it to that memory block. As seen in Fig. 2, there are no combinational logic gates. Thus, the super-block D-cubes of Example A may be defined simply with the primitive flip-flop

D-cubes which were identified in the previous section. The pertinent primitive D-cubes and α condition cubes for each flip-flop in Example A are listed in Table III. These will now be used to define D-chains for the sequential machine in Example A.

As can be seen in Table III, each D-cube or α -cube has a unique shorthand name, $F_1\sigma$, $\bar{F}_2\alpha$, etc. which refers to the final state of the memory variable or machine output variable generated by the cube for its super-block and a σ, ρ, τ, or α state control condition with which it is associated. These names will be used to identify the primitive super-block D-cubes which form a sequential D-chain. As explained earlier, the sequential D-chain is constructed by chaining a machine output D-cube to a D-cube which provides its D-term input coordinate and so on until the input D-coordinate term is a controllable machine input or a memory element already in the chain. Using this technique on Example A (which can be applied in a straightforward manner since no asynchronous coupling exists via unclocked inputs), one obtains the sequential D-chains:

1) $F_1\sigma$

2) $F_1\tau$, $T_1\tau$

3) $F_1\tau$, $T_1\tau$

4) $\bar{F}_1\rho$

5) $\bar{F}_1\tau$, $T_1\tau$.

These sequential D-chains define propagation paths and sensitizing conditions for Example A in terms of local flip-flop state control conditions. The first one of these D-chains, $F_1\sigma$, will now be converted to an initializing and exercising sequence in order to illustrate the use of the \$ calculus.

In generating the exercising and initializing sequences for a D-chain, a calculus rule which must be followed is that no memory variable may be left with a 1/0, 0/1, D/1, or D/0 state. If this occurs, additional cubes as indicated by the nature of the state must be combined with the original one to completely define that step of the sequence. This sort of circumstance occurs in the D-chain, $F_1\sigma$. As seen in Table III, the T_1 coordinate of F_1 is a 1/0 state. Thus a cube which toggles T_1 to a final state which is false is needed to completely define the last step of the exercising sequence. This is possible if $\bar{T}_1\tau$ is combined with $F_1\sigma$ in order to completely define the final state of the exercise sequence $F_1 = 1$, $T_1 = 0$. Using the \$ calculus, one obtains

$$F_1\sigma\ \$\bar{T}_1\tau\ = \bar{t}\ T\ D\ 0\ D/0\ 0 \tag{7}$$

which defines the previous state of the machine and the input signal states which caused state transition. In this case, using the exercise previous state interpretation given in the preceding section (7) indicates that $F_1 = 0$, $T_1 = 1$ in the previous machine state and $J = 1$, $K = 0$, $C = 1/0$, and $S = 0$ caused the state transition. Since there are no more D-cubes in the chain, the previous machine state defined by (7) is the start of the initializing sequence. Commencing the initializing sequence, various cubes for $F_1 = 0$ and $T_1 = 1$ from Table III may then be combined using the \$ operator; however, only $\bar{F}_1\alpha_2$ and $T_1\tau$ yield a non-* result:

$$\bar{F}_1\alpha_2\ \$T_1\sigma\ = \bar{A}\ \bar{t}\ 0\,0\,X\,d \tag{8}$$

which indicates that the previous machine state is $F_1 = 0$ and $T_1 = X$ and that the input signal states responsible for state transition are $J = 0$, $K = 0$, $C = X$, and $S = 1$. To obtain the next previous machine state, if one selects the cube $\bar{F}_1\rho$, the need for $T_1 = 0$ is indicated by the 1/0 term for T_1. Therefore, also selecting $\bar{T}_1\tau$ for $T_1 = 0$ as an additional term, one obtains

$$\bar{F}_1\rho\ \$\bar{T}_1\tau\ = t\,T\,0\,D\,D/0\,0 \tag{9}$$

which indicates $F_1 = X$, $T_1 = 1$ previously and $J = 0$, $K = 1$, $C = 1/0$ and $S = 0$. Finally, selecting $T_1\sigma$ for $T_1 = 1$,

$$T_1\sigma\ = X\,\bar{t}\,X\,X\,X\,d \tag{10}$$

indicates $F_1 = X$, $T_1 = X$ previously and $J = K = C = X$ and $S = 1$ to cause transition. This is the last step of the initialize sequence because the next previous state is a complete don't care for the memory variables. If all the steps are now written together, the complete test for sequential D-chain F_1 is

$$\begin{array}{l} \left.\begin{array}{ll} t = 1 & X\,X\,X\,X\,X\quad 1 \\ t = 2 & X\,1\,0\,1\,1/0\,0 \\ t = 3 & 0\,0\,0\,0\,X\quad 0 \end{array}\right\}\ \text{INITIALIZE} \\ \left.\begin{array}{ll} t = 4 & 0\,1\,D\,0\,1/0\,1 \\ t = 5 & 1\,0 \end{array}\right\}\ \text{EXERCISE.} \end{array} \tag{11}$$

The test sequence described in (11) may be used to test the flip-flop with output F_1 for F_1 and J stuck at one or zero and to detect faulty sensitizing conditions. Generally, if the controllable input D-term for a chain is appropriately set ($D = 1$ or $\bar{D} = 0$) the test (TYPE A) sequence will check for D-terms in the chain stuck at zero and $\bar{D}$-terms stuck at one. Alternatively with the input $D = 0$ or $\bar{D} = 1$ the test (TYPE B) will check for $\bar{D}$-terms stuck at one and $\bar{D}$ stuck at zero. If one does this for the test sequence in (11), it will test F_1 and J, but be somewhat more powerful than what is minimally sufficient. This occurs because the memory variables on the path $F_1\sigma$ are totally observable which corresponds to the degenerate case mentioned in the previous section. The D-chain $F_1\sigma$ is nevertheless useful for illustrating the sequential path sensitizing method and the \$ calculus.

A more complicated sequential machine, Example B shown in Fig. 3, will now be considered. An appropriate covering set of D-cubes and α-cubes for the machine for the machine are given in Table IV. These cubes are defined for the super logic blocks of the machine. In this super block set, the combinational logic subset at the left of Fig. 3 has been absorbed into a super flip-flop with output F_1, the logic gate with machine output G is taken as a super output block, and the flip-flops with outputs F_0 and F_2 are super flip-flops by themselves with no combinational logic subsets. Some of the sequential D-chains which one can generate for the Example B machine using the method previously described are:

1) $G\sigma_1,\ F_2\sigma_1,\ F_0\sigma_1$

2) $G\sigma_1,\ F_2\ \sigma_2,\ F_1\sigma_1$

3) $G\sigma_1,\ F_2\ \sigma_2,\ F_1\sigma_2$

4) $G\sigma_1,\ F_2\ \sigma_2,\ F_1\sigma_3$

5) $G\sigma_1,\ F_2\ \sigma_2,\ F_1\sigma_4,\ F_2\ \sigma_1$

6) $G\sigma_1,\ F_2\ \sigma_2,\ F_1\sigma_5,\ F_2\ \sigma_2$.

Several of these will now be used to generate test sequences in order to illustrate several facets of sequential D-chains and the $ calculus which have not yet been demonstrated. The first of these is $G\,\sigma_1$, $F_2\sigma_1$, $F_0\sigma_1$ which is a set of super block D-cubes asynchronously coupled via unclocked inputs. To handle this chain, a second calculus rule must be defined which concerns asynchronously coupled blocks: There must be no d or $\bar{d}$ terms in the memory variable coordinates of a resultant cube. Cubes which are coupled via a d or $\bar{d}$ input must be combined with the $ calculus. For the set of cubes, $G\sigma_1$, $F_2\sigma_1$, $F_0\sigma_1$, one does not obtain a complete definition of the exercise sequence final state (with no d or $\bar{d}$ input from any memory variable) unless all the super block D-cubes are combined to yield

$$G\sigma_1\ \$F_2\,\sigma_1\ \$F_0\,\sigma_1\ =\ \bar{t}\bar{t}\,\mathrm{X}\,\bar{t}\,\mathrm{d\,XXX10}. \tag{12}$$

The resultant cube in (12) indicates that $G = F_0 = F_2 = 0$ in the previous state and that A = E = 1 caused transition. Since there are no more D-cubes in the chain this previous state is part of the initializing sequence. Continuing, then, one obtains the initializing sequence

$$\begin{aligned} G = F_0 = F_2 = 0&:\ \bar{G}\rho_1\ \$F_2\rho\,\$F_0\alpha = \mathrm{t}\bar{\mathrm{A}}\bar{\mathrm{D}}\mathrm{t\,0X1/0X10} \\ F_0 = F_1 = 0&:\ \bar{F}_0\alpha\,\$\bar{F}_1\,\rho_1 = \mathrm{X}\bar{\mathrm{A}}\mathrm{t\,X0}\bar{\mathrm{D}}\mathrm{1/01X0} \\ F_0 = 0&:\ \bar{F}_0\,\rho_1 = \mathrm{Xt\,XX0XXXXd}. \end{aligned} \tag{13}$$

Thus, the complete test is

$$\begin{array}{l} \left.\begin{array}{lll} t=1 & \mathrm{XXXX} & \mathrm{0XX\quad XX1} \\ t=2 & \mathrm{X0XX} & \mathrm{001/01X0} \\ t=3 & \mathrm{X00X} & \mathrm{0X1/0X10} \end{array}\right\}\ \text{INITIALIZE} \\ \left.\begin{array}{lll} t=4 & \mathrm{00X0} & \mathrm{dXX\quad X10} \\ t=5 & \mathrm{11X1} & \end{array}\right\}\ \text{EXERCISE} \end{array} \tag{14}$$

which demonstrates how an asynchronously coupled set of cubes in a sequential D-chain are handled.

Another interesting sequential D-chain in Example B is $G\sigma_1$, $F_2\ \sigma_2$, $F_1\sigma_4$ which corresponds to a synchronous logical loop. In this chain of cubes it can be seen that the chain terminates where it connects back upon itself. If one applies the $ calculus, one obtains

$$\begin{aligned} &G = F_2 = 1,\ F_0 = 0:\ G\sigma_1\ \$F_2\sigma_2\ \$\bar{F}_0\alpha = \\ &\qquad \bar{\mathrm{t}}\,\bar{\mathrm{A}}\,\mathrm{D}\,\bar{\mathrm{t}}\ \mathrm{0X1/0X10} \\ &G = F_2 = F_0 = 0,\ F_1 = 1:\ \bar{G}\rho_1\ \$\bar{F}_0\rho_1\ \$F_1\sigma_4\ \$\bar{F}_2\rho = \\ &\qquad \mathrm{t}\bar{\mathrm{A}}\bar{\mathrm{T}}\mathrm{T\,0X1/001d} \end{aligned} \tag{15}$$

for the exercising sequence and

$$G = 1,\ F_0 = F_1 = 0,\ F_2 = 1: G\sigma_1\ \$\bar{F}_0\rho_1\ \$\bar{F}_1\rho_1\ \$F_2\sigma_2 =$$

$$\bar{\mathrm{t}}\,\bar{\mathrm{A}}\,\mathrm{T}\,\bar{\mathrm{t}}\ \mathrm{0}\bar{\mathrm{D}}\mathrm{1/011d}$$

$$F_0 = 0,\ F_1 = 1:\ \bar{F}_0\alpha\$F_1\sigma_1 = \mathrm{XAtA011/01X0}$$

$$F_0 = 0:\ \bar{F}_0\rho_1 = \mathrm{XtXX0XXXX1}$$

for the initializing sequence which defines a TYPE A test

$$\begin{array}{l} \left.\begin{array}{lll} t=1 & \mathrm{XXXX} & \mathrm{0XX\quad XX1} \\ t=2 & \mathrm{X0XX} & \mathrm{011/01X0} \\ t=3 & \mathrm{X01X} & \mathrm{001/0111} \end{array}\right\}\ \text{INITIALIZE} \\ \left.\begin{array}{lll} t=4 & \mathrm{1001} & \mathrm{0X1/0011} \\ t=5 & \mathrm{0010} & \mathrm{0X1/0X10} \\ t=6 & \mathrm{10X1} & \end{array}\right\}\ \text{EXERCISE} \end{array} \tag{16}$$

It may be noted that a TYPE B test for the same path would require generation of an initializing sequence for $G = F_0 = F_1 = F_2 = 0$.

The last sequential D-chain which will be discussed is the chain $G\sigma_1$, $F_2\sigma_2$, $F_1\sigma_1$. It has been stated in the previous section that a diagnosable path is one which does not allow state changes to enter a path simultaneously via more than one route during the execution of the exercising sequence. The sequential D-chain $G\sigma_1$, $F_2\sigma_2$, $F_1\sigma_1$ has a set of cubes which violate this rule for a TYPE A test. The obvious cause of circumstance is that there exists a cube $F_2\sigma_1$ with input variable state requirements similar to that of $F_1\sigma_1$. Since these cubes are defined in the equivalent state transition model only locally for each flip-flop, the \$ calculus as it is will not detect the problem. Thus, one must have a list of D-cubes with similar state transition requirements or simulate a test in order to detect such a sequential D-chain. Though such chains cannot be diagnosed in terms of a test which is strictly TYPE A or TYPE B, it is possible to devise a test which is composed of TYPE A and TYPE B segments[10]. Alternatively, one may also add observable test points at appropriate places. In Example B, for instance, one may add a test point at F_1 to break the chain $G\sigma_1$, $F_2\sigma_2$, $F_1\sigma_1$. The part, $F_1\sigma_1$, then is a sequential D-chain by itself and the cubes $G\sigma_1$, $F_2\sigma_2$ otherwise are included in a chain $G\sigma_1$, $F_2\sigma_2$, $F_1\sigma_1$.

CONCLUSIONS

In the foregoing, the path sensitizing method, well known for testing combinational logic nets, was extended to the case of the sequential machine. This extension is discussed in terms of D-chain test theory. Initially embryonic tests and primitive D-cubes for a flip-flop are developed. A method for chaining these to each other directly and via combinational D-chains is then presented. The central point of this discussion is a description of the "sequential D-chain", a set of sequentially executed D-cubes which define a "propagation path" in the sequential machine and its sensitizing conditions in terms of state transition conditions which are local to each memory element on the path. A calculus is then presented which allows one to determine machine level state transition information from local memory element state transition information for each sequential D-chain.

It is estimated that the technique presented here should permit diagnosis of large sequential machine structures which previously have been difficult or impossible to handle with the complete machine "distinguishing sequence - equivalence class partitioning" methods of diagnosis. The method is an extremely programmable one for automated generation of test sequences. Such a computer program has recently been implemented by the author and his associates.

REFERENCES

1 E. F. Moore, "Gedanken - experiments on sequential machines," Automata Studies, Princeton, New Jersey: Princeton University Press, pp. 129-153, 1956.

2 J. D. Brule et al., "Diagnosis of equipment failures," IRE Trans. on Reliability and Quality Control, vol. RQC-9, pp. 23-24, April 1960.

3 S. Seshu and D. N. Freeman, "The diagnosis of asynchronous sequential switching systems," IRE Trans. on Electronic Computers, vol. EC-11, pp. 459-465, August 1962.

4 S. Seshu, "On an improved diagnosis program," IEEE Trans. on Electronic Computers, vol. EC-14, pp. 76-79, February 1965.

5 F. C. Hennie, "Fault detecting experiments for sequential circuits," Proc. 5th Symposium on Switching Theory and Logic Design, pp. 95-110, 1964.

6 C. R. Kime, "An organization for checking experiments on sequential circuits," IEEE Trans. on Electronic Computers, vol. EC-15, pp. 113-115, February 1966.

7 D. B. Armstrong, "On finding a nearly minimal set of fault detection tests for combinational logic nets," IEEE Trans. on Electronic Computers, vol. EC-15, pp. 66-73, February 1966.

8 J. P. Roth, "Diagnosis of automata failures: A calculus and a method," IBM Journal of Research and Development, vol. 10, pp. 278-291, July 1966.

9 F. Sellers, M. Hsiao, and L. Bearnson, "Analyzing errors with the boolean difference," Digest, First Annual IEEE Computers Conference, pp. 6-9, Septebmer 1967.

10 T. A. Kriz, "Machine identification concepts of path sensitizing fault diagnosis," TENTH ANNUAL SYMPOSIUM ON SWITCHING AND AUTOMATA THEORY, Waterloo, Canada, October 1969.

11 A.Gill, An Introduction to the Theory of Finite State Machines, New York: McGraw-Hill Book Company, pp. 88-90, 1962.

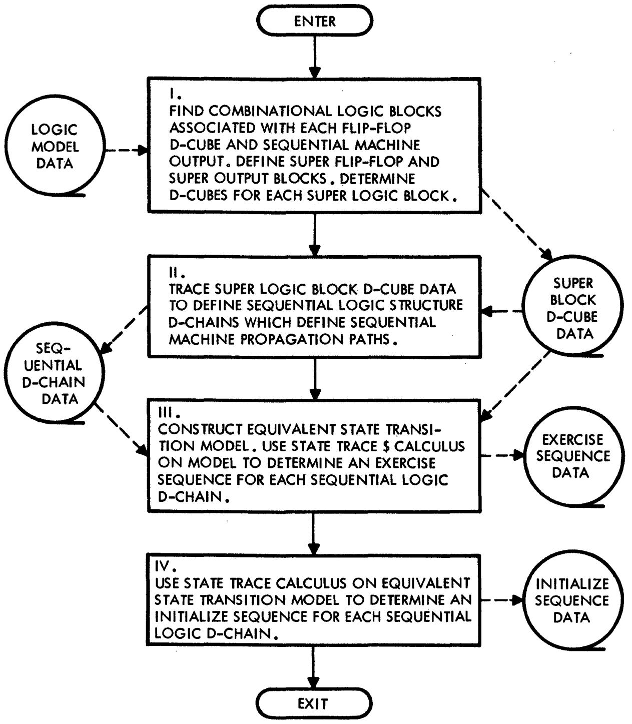

ENTER
LOGIC MODEL DATA
I. FIND COMBINATIONAL LOGIC BLOCKS ASSOCIATED WITH EACH FLIP-FLOP D-CUBE AND SEQUENTIAL MACHINE OUTPUT. DEFINE SUPER FLIP-FLOP AND SUPER OUTPUT BLOCKS. DETERMINE D-CUBES FOR EACH SUPER LOGIC BLOCK.
II. TRACE SUPER LOGIC BLOCK D-CUBE DATA TO DEFINE SEQUENTIAL LOGIC STRUCTURE D-CHAINS WHICH DEFINE SEQUENTIAL MACHINE PROPAGATION PATHS.
SUPER BLOCK D-CUBE DATA
SEQ-UENTIAL D-CHAIN DATA
III. CONSTRUCT EQUIVALENT STATE TRANSITION MODEL. USE STATE TRACE $ CALCULUS ON MODEL TO DETERMINE AN EXERCISE SEQUENCE FOR EACH SEQUENTIAL LOGIC D-CHAIN.
EXERCISE SEQUENCE DATA
IV. USE STATE TRACE CALCULUS ON EQUIVALENT STATE TRANSITION MODEL TO DETERMINE AN INITIALIZE SEQUENCE FOR EACH SEQUENTIAL LOGIC D-CHAIN.
INITIALIZE SEQUENCE DATA
EXIT

Figure 1. Sequential Path Sensitizing Algorithm Outline.

TABLE I. EMBRYONIC TESTS FOR A J-K/R-S FLIP-FLOP

INHIBIT D-CUBES	TRIGGER D-CUBES	CONTROL CONDITION	
F S R J K C	F S R J K C		
$\bar{D}$ $\bar{D}$ 0 X X 0	D D 0 X X 0	σ, FIRST TERM	F' = 0 INITIALLY
$\bar{D}$ $\bar{D}$ 0 0 0 X	D D 0 0 0 X		
$\bar{D}$ $\bar{D}$ 0 X X 1	D D 0 X X 1		
	plus others		
$\bar{D}$ 0 0 $\bar{D}$ 0 1/0	D 0 0 D 0 1/0	σ, SECOND TERM	
$\bar{D}$ 0 0 1 0 $\bar{D}$/0	D 0 0 1 0 D/0		
$\bar{D}$ 0 0 1 1 $\bar{D}$/0	D 0 0 1 1 D/0	τ	
D 0 $\bar{D}$ X X 0	$\bar{D}$ 0 D X X 0	ρ, FIRST TERM	F = 1 INITIALLY
D 0 $\bar{D}$ 0 0 X	$\bar{D}$ 0 D 0 0 X		
D 0 $\bar{D}$ X X 1	$\bar{D}$ 0 D X X 1		
	plus others		
D 0 0 0 $\bar{D}$ 1/0	$\bar{D}$ 0 0 0 D 1/0	ρ, SECOND TERM	
D 0 0 0 1 $\bar{D}$/0	$\bar{D}$ 0 0 0 1 D/0		
D 0 0 1 1 $\bar{D}$/0	$\bar{D}$ 0 0 1 1 D/0	τ	

TABLE II. DEFINITION OF $ OPERATOR

	$\bar{D}$,	D,		D/0	$\bar{D}$/1								
$	0	1	X	1/0	0/1	T	$\bar{T}$	t	$\bar{t}$	d	$\bar{d}$	A	$\bar{A}$
$\bar{D}$, 0	0	*	0	*	0/1	*	$\bar{T}$	$\bar{A}$	$\bar{T}$	*	$\bar{d}$	*	$\bar{A}$
D, 1	*	1	1	1/0	*	T	*	T	A	d	*	A	*
X	0	1	X	1/0	0/1	T	$\bar{T}$	t	$\bar{t}$	d	$\bar{d}$	A	$\bar{A}$
D/0, 1/0	*	1/0	1/0	1/0	*	T	*	T	*	*	*	*	*
$\bar{D}$/1, 0/1	0/1	*	0/1	*	0/1	*	$\bar{T}$	*	$\bar{T}$	*	*	*	*
T	*	T	T	T	*	-	-	-	-	*	T	-	-
$\bar{T}$	$\bar{T}$	*	$\bar{T}$	*	$\bar{T}$	-	-	-	-	$\bar{T}$	*	-	-
t	$\bar{A}$	T	t	T	*	-	-	-	-	*	t	-	-
$\bar{t}$	$\bar{T}$	A	$\bar{t}$	*	$\bar{T}$	-	-	-	-	$\bar{t}$	*	-	-
d	*	d	d	*	*	*	$\bar{T}$	*	$\bar{t}$	d	*	A	*
$\bar{d}$	$\bar{d}$	*	$\bar{d}$	*	*	T	*	t	*	*	$\bar{d}$	*	$\bar{A}$
A	*	A	A	*	*	-	-	-	-	A	*	-	-
$\bar{A}$	$\bar{A}$	*	$\bar{A}$	*	*	-	-	-	-	*	A	-	-

Inputs:

1 = true state
0 = false state
X = don't care
1/0 = true-to-false transition
0/1 = false-to-true transition
$\bar{D}$, D, D/0, $\bar{D}$/1 = D-states
d, $\bar{d}$ = asynchronous D-input

Outputs:

$\bar{t}$ = true final state
t = false final state
$\bar{T}$ = 0/1 toggle
T = 1/0 toggle
A = true at rest
$\bar{A}$ = false at rest
* = prohibited state

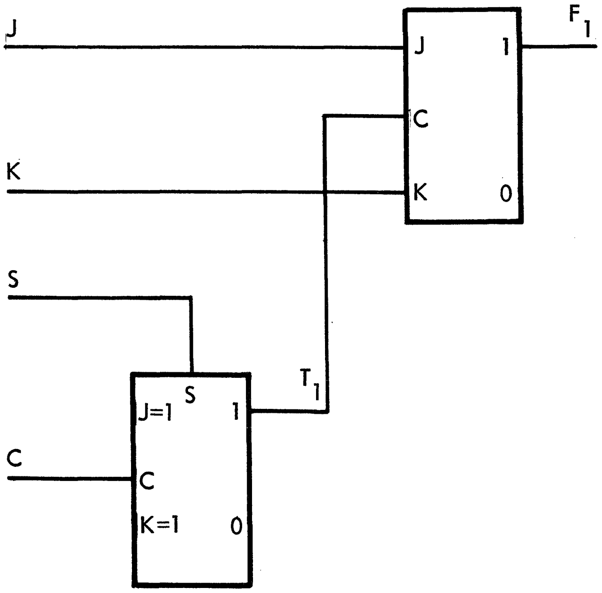

J
F1
J
1
C
K
K
0
S
T1
J=1
S
1
C
C
K=1
0

Figure 2. Example A.

TABLE III. EXAMPLE A STATE TRANSITION CUBES

FINAL STATE		STATE TRANSITION CUBES						CUBE NAME
F_1	T_1	F_1	T_1	J	K	C	S	
1		$\bar{t}$	1/0	D	0	X	X	$F_1\sigma$
1		T	D/0	1	1	X	X	$F_1\tau$
1		A	0/1	X	X	X	X	$F_1\alpha_1$
1		A	X	0	0	X	X	$F_1\alpha_2$
0		t	1/0	0	D	X	X	$\bar{F}_1\rho$
0		T	D/0	1	1	X	X	$\bar{F}_1\tau$
0		$\bar{A}$	0/1	X	X	X	X	$\bar{F}_1\alpha_1$
0		$\bar{A}$	X	0	0	X	X	$\bar{F}_1\alpha_2$
	1	X	$\bar{T}$	X	X	D/0	0	$T_1\tau$
	1	X	$\bar{t}$	X	X	X	d	$T_1\sigma$
	1	X	A	X	X	0/1	0	$T_1\alpha$
	0	X	T	X	X	D/0	0	$\bar{T}_1\tau$
	0	X	A	X	X	0/1	0	$\bar{T}_1\alpha$

TABLE IV. EXAMPLE B STATE TRANSITION CUBES

STATE TRANSITION CUBES										CUBE NAME
G	F_0	F_1	F_2	A	B	C	D	E	H	
$\bar{t}$	X	X	d	X	X	X	X	1	X	$G\sigma_1$
$\bar{t}$	X	X	1	X	X	X	X	d	X	$G\sigma_2$
t	X	X	$\bar{d}$	X	X	X	X	1	X	$\bar{G}\rho_1$
t	X	X	1	X	X	X	X	$\bar{d}$	X	$\bar{G}\rho_2$
X	d	X	$\bar{t}$	X	X	X	X	X	X	$F_2\sigma_1$
X	0	D	$\bar{t}$	X	X	1/0	X	X	X	$F_2\sigma_2$
X	0	$\bar{D}$	t	X	X	1/0	X	X	X	$\bar{F}_2\rho$
X	0	X	A	X	X	0	X	X	X	$F_2\alpha$
X	0	X	$\bar{A}$	X	X	0	X	X	X	$\bar{F}_2\alpha$
X	d	$\bar{t}$	X	X	X	X	X	X	X	$F_1\sigma_1$
X	0	t	X	X	D	1/0	1	X	X	$F_1\sigma_2$
X	0	$\bar{t}$	X	X	1	D/0	1	X	X	$F_1\sigma_3$
X	0	$\bar{t}$	D	X	X	1/0	0	X	X	$F_1\sigma_4$
X	0	$\bar{t}$	1	X	0	1/0	$\bar{D}$	X	X	$F_1\sigma_5$
X	0	A	X	X	X	0	X	X	X	$F_1\alpha$
X	0	t	X	X	$\bar{D}$	1/0	1	X	X	$\bar{F}_1\rho_1$
X	0	t	$\bar{D}$	X	X	1/0	0	X	X	$\bar{F}_1\rho_2$
X	0	t	0	X	1	1/0	$\bar{D}$	X	X	$\bar{F}_1\rho_3$
X	0	A	X	X	X	0	X	X	X	$\bar{F}_1\alpha$
X	$\bar{t}$	X	X	d	X	X	X	X	0	$F_0\sigma_1$
X	A	X	X	0	X	X	X	X	0	$F_0\alpha$
X	t	X	X	0	X	X	X	X	d	$\bar{F}_0\rho_1$
X	A	X	X	0	X	X	X	X	0	$\bar{F}_0\alpha$

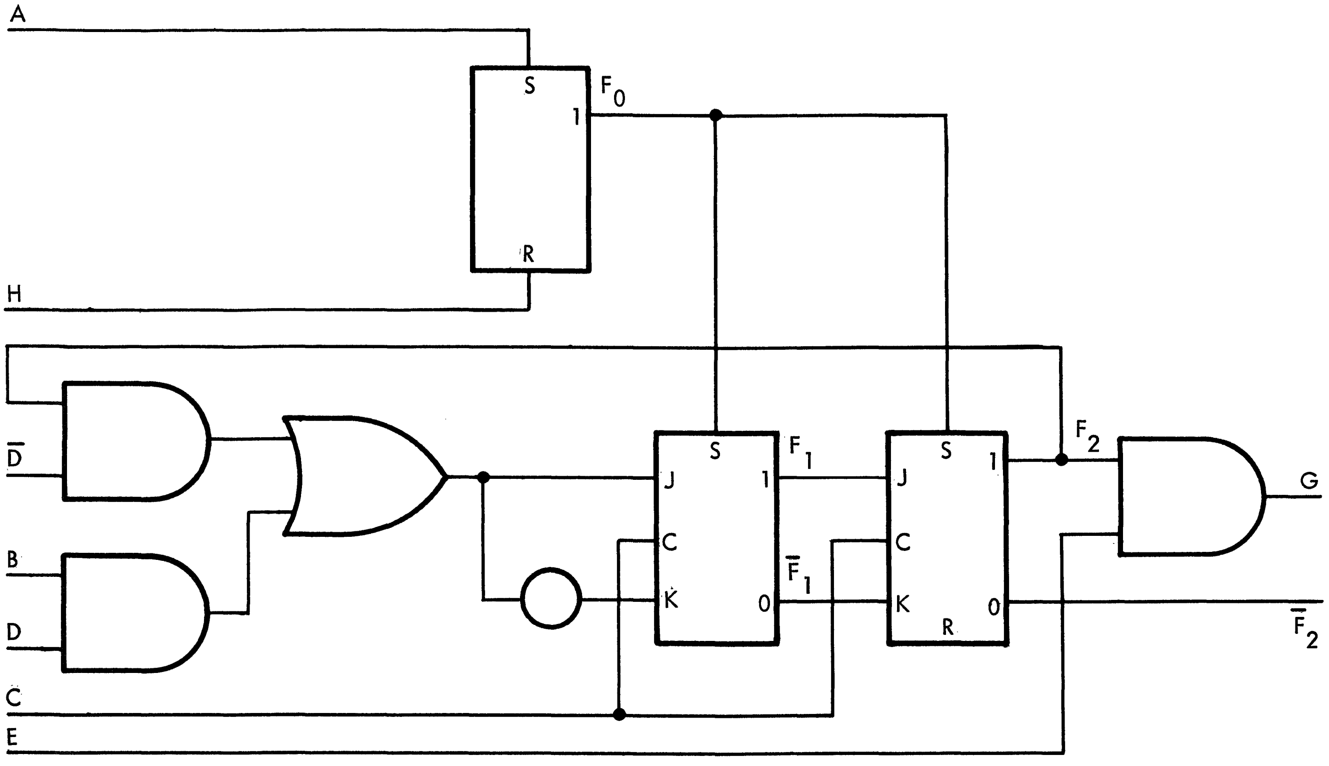


Figure 3. Example B.

ELECTROLUMINESCENT FLAT PANEL DISPLAY

M. Yoshiyama, H. Kawarada, and T. Sato
Wireless Research Laboratory, Matsushita Electric Industrial Co., Ltd.
1006 Kadoma, Osaka, 571, Japan

ABSTRACT

A 52,900-element electroluminescent (EL) flat panel display, based on the improved EL image display system, has been made by using a powder type EL matrix panel with a 13-inch diagonal. The experimental display can reproduce both static and moving images with half-tone and characters on the panel, from 525-line standard television signals in real time. The reproduced images, with highlight brightness of about 1 Foot-Lambert and contrast ratio of 5:1 to 3:1, have gray-scale capability of 4 or 5 steps with uniform brightness over the whole panel. The resolution of the images is better than that of an inexpensive home video tape recorder. The cross effect of the panel is almost entirely suppressed by the blanking pulses. Power dissipation of the display is about 100 watts. The display is designed to operate with TV signals, but can be used for other applications such as character displays, when modified.

INTRODUCTION

The flat panel display is a device that is considerably less thick than it is high or wide. It converts various input information into visual images. Flat panel displays are conveniently classified into two groups: light generating and light reflecting or transmitting. The former includes, for example, Plasma displays and electroluminescent (EL) displays. The liquid crystal display is an example of the latter. Although the EL display has been used principally in wall-mounted type television units, it is now being applied to other areas such as numeric or computer-generated character displays, to satisfy both industrial and military requirements.

EL displays can be divided roughly into the powder or intrinsic type and the injection type. Although the emphasis of development has shifted almost completely to injection types, powder types still have a high potential for relatively large displays. Injection type EL displays, using light emitting diodes of III-V compounds, are available for numerical displays. At present, it is both difficult and prohibitively expensive to fabricate large display panels by using single crystal substrate. Powder type EL panels can be fabricated on a glass plate by spraying or silk-screen techniques, thereby permitting large uniform panels to be fabricated at relatively low cost. However, a further investigation is necessary in order to obtain satisfactory EL phosphor materials with higher light output, higher efficiency and longer life.

Other technical problems encountered in displays built with EL matrix panels are: how to scan the panel in response to information signals and how to suppress the cross effect relating to contrast of images. The EL panels can be scanned by several known methods, such as classical X-Y matrix, transfluxer [1], ferroelectric, [2,3,4,5] and piezoelectric control types.[6]

A display for TV images must satisfy three requirements: (1) capability to scan for a large number of picture elements, (2) high scanning speed, and (3) reproduction of moving half-tone images. Therefore, constructing a display for TV imagery is more difficult than that for any other class of images.

This paper describes a flat panel display which utilizes powder type EL matrix panel in practically large scale. A 52,900-element EL matrix panel, its scanning system, an experimental display of a TV image and its performance are discussed.

EL MATRIX PANEL

Fig. 1 shows a powder type EL matrix panel in crossed-grid structure, which includes a layer having EL phosphor materials embedded in a plastic binder.

The 52,900-element EL matrix panel shown in Fig. 2 is prepared on a glass plate by integrating the following multiple layers; transparent vertical electrodes, an EL phosphor layer, a non-linear impedance layer, horizontal electrodes, and an epoxy protective coating. The EL phosphor layer is basically sandwiched between 230 vertical and 230 horizontal electrodes. The 52,900 EL elements are formed at their crossing points as shown in Fig. 1. The transparent 230 vertical electrodes in a strip form are prepared by photoetching tin oxide coating on the glass plate; each strip is 1.0 mm in width and 0.2 mm in spacing. The EL phosphor layer in about 35 μm thickness is composed of ZnS (Cu, Al) phosphor embedded in cyano-ethylated cellulose. It is superimposed uniformly on the 230 vertical electrodes by a silk-screen technique. The EL phosphor has a whitish green emission. The non-linear impedance layer in about 10 μm thickness is superimposed on the EL phosphor layer in order to increase the non-linear characteristic of the panel. This layer is composed of barium titanate powder embedded in cyano-ethylated cellulose, and it also acts as a reflecting layer of emitted light. The 230 horizontal electrodes are made of evaporated-aluminum strips; each strip is 0.75 mm in width and 0.15 mm in spacing. Each EL element measures 0.75 mm by 1.0 mm. The total display area of the panel is 207 mm by 276 mm, which is almost equal to that of a 13-inch cathode ray tube (CRT). The aspect ratio of the display area is 4:3.

The glass plate is mounted on an insulating base board. The horizontal and vertical electrode strips are bonded to printed connecting boards fixed to the base board. They can be connected easily and reliably to the driving circuits.

Light output of the EL element usually increases with increasing amplitude and repetition frequency of applied pulse. It has a waveform similar to the differential waveform of applied pulse; the waveform of light output shows a rapid increase followed by an exponential decay at each rise and fall of pulse voltage. Thus, the light output of the EL element cannot be easily increased only by lengthening pulse duration.

The luminance of the EL element is not a linear function of pulse voltage, and can be approximately expressed by the following equation in a limited voltage range;

$$L = kV^t, \qquad (1)$$

where L is luminance, k a constant of proportionality, V an applied pulse voltage, and t a non-linear coefficient. For a typical EL phosphor layer composed of ZnS (Cu, Al) phosphor embedded in cyano-ethylated cellulose binder, the t is about 3.5, as shown in Fig. 3. This value of t is not sufficient to reproduce images with high contrast. The coefficient t can be enlarged up to 4 by adding a non-linear impedance layer.

Vecht, Werring, and Smith[7] recently reported on a DC-EL phosphor layer composed of copper coated ZnS (Mn, Cu, Cl) powder. This material has a high t of about 8 and will be suitable for character display panels which need high contrast.

SCANNING SYSTEM OF EL MATRIX PANEL

The EL display on which we report is operated by a scanning method of X-Y matrix control type, and is characterized by (1) application of sequential line-by-line (line-at-a-time) scanning, (2) brightness control by pulse amplitude modulation, and (3) suppression of the cross effect by blanking pulses.

Although sequential element-by-element (element-at-a-time) scanning method is used for that of the CRT in a conventional TV receiver, its application to the EL matrix panel with a large number of EL elements results in low brightness due to short exciting time of the EL elements. Instead of the above scanning method, the EL display uses the sequential line-by-line scanning method by a tapped video delay line. This method not only simplifies the system configuration by eliminating the need for horizontal scanning, but also increases the brightness of the EL elements.

Fig. 4 shows a block diagram of an EL display of TV image based on an improved EL image display system. In order to scan an EL matrix panel, a selecting pulse is applied to the horizontal electrode strip to be scanned. Simultaneously, corresponding multi-channel video signals are applied to brightness control circuits. These circuits are connected to vertical electrode strips to control brightness of all EL elements along the selected horizontal electrode. A video signal for one horizontal line is converted into multi-channel video signals by a tapped video delay line, having a total delay time of one effective horizontal period. The video signal must receive a gamma correction to compensate for the non-linear characteristic of the panel. When a negative selecting pulse having a given amplitude is applied to the selected horizontal electrode strip, the multi-channel video signal modulates the collector resistance of output transistor in each of brightness control circuits. Consequently, the signal controls the voltage across the EL element. When the collector resistance is a minimum, the EL element is supplied with a major part of selecting pulse voltage to produce maximum brightness.

The horizontal electrode driving circuit has a set of pulse generators designated as HjPG in Fig. 4. They supply a selecting pulse to the selected horizontal electrode strip. At the same time, blanking pulses are supplied to the remaining electrode strips at every horizontal period of the video signal. The blanking pulses are in phase with the selecting pulse.

This system is designed to suppress the cross effect by the blanking pulses. The technique eliminates the need of diode at each EL element location of the panel. Use of these diodes makes it difficult to fabricate a panel with high quality and reliability because of its complicated structure.

Suppression of the cross effect by the use of blanking pulses in the sequential line-by-line scanning system can be explained with a special pattern shown in Fig. 5(a). Suppose that a number of electrode strips in the horizontal or vertical direction is equal to n, and that each of the EL elements at their crossing point has an equal capacitance C. The display of the pattern shown in Fig. 5(a) corresponds to an equivalent circuit of Fig. 5(b). When a selecting pulse V is applied only to the selected horizontal electrode strip, the voltage across the EL elements of the dark portion becomes a value of (n-1/n+1)V. Consequently, the discrimination between the bright and the dark portion of the image will almost disappear for a relatively large value of n. When blanking pulses having an amplitude of V/a (a>1) in phase with the selecting pulse are further applied to the remaining horizontal electrode strips at the same time, the voltage V_D across the EL elements in the dark portion can be calculated from the equivalent circuit shown in Fig. 5(b);

$$V_D = \frac{n-1}{n} \cdot (1-\frac{1}{a})V. \qquad (2)$$

For relatively large values of n, the voltage will become approximately equal to a value of

$$V_C = (1-\frac{1}{a})V. \qquad (3)$$

The integrated light output of the EL elements during the scanning depends on the light caused not only by the selecting pulse, but also by the blanking pulses, though the light by the blanking pulses does not have a strong effect. The constant of proportionality k and the non-linear coefficient t in equation (1) are functions of repetition frequency of applied pulse. For simplicity, the luminance per one pulse is expressed in a form of equation (1) as follows:

$$L_1 = k_1 V^{t_1}, \qquad (4)$$

$$L_2 = k_2 V^{t_2}, \qquad (5)$$

where suffixes 1 and 2 are for the selecting pulse and the blanking pulse, respectively.

When the pattern shown in Fig. 5(a) is displayed on the whole panel by this scanning system, the contrast ratio (C.R.) of the bright and the dark portion of the image can be calculated from equations (3), (4) and (5);

$$\text{C.R.} = \frac{1+(n-1)\frac{k_2}{k_1}(\frac{1}{a})^{t_2} V^{t_2-t_1}}{(1-\frac{1}{a})^{t_1}} \qquad (6)$$

For another pattern which is bright at the upper half and dark at the lower half, the contrast ratio can also be calculated in the same manner;

$$\text{C.R.} = \frac{1+(\frac{n}{2}-1)\frac{k_2}{k_1} \cdot (\frac{1}{a})^{t_2} V^{t_2-t_1}}{(1-\frac{1}{a})^{t_1} + \frac{n}{2} \cdot \frac{k_2}{k_1}(\frac{1}{a})^{t_2} V^{t_2-t_1}} \qquad (7)$$

It is easy to calculate the contrast ratio for simple patterns, but it is difficult for complex ones. The contrast ratio and the value of "a" at which the maximum ratio is obtained, change with each pattern to be displayed. As a result of experiments, the "a" of about 3 produces practically a satisfactory contrast for all patterns. For the non-linear coefficient t, the contrast ratio increases with an increase in the value of "t".

EXPERIMENTAL 52,900-ELEMENT DISPLAY OF TV IMAGE AND ITS PERFORMANCE

1. Outline

An experimental EL image display system, schematically represented in Fig. 4, has been made. The EL matrix panel has 52,900 EL elements and a total picture area of 13 inches diagonal as described above. The experimental display is designed to operate with 525-line standard TV signal at 60 fields. It has junction terminals for a video tape recorder and for a closed-circuited TV camera. Since the number of horizontal electrodes of the EL matrix panel is 230 lines, which is approximately equal to a half number of effective scanning lines in television frame, two field images are successively displayed without interlace on the same horizontal lines. The amplitude of the selecting pulse is about 300 volts in negative polarity. The blanking pulses in phase with the selecting pulse have an amplitude of about one-third that of the selecting pulse.

The experimental display is composed of a 52,900-element EL matrix panel, two speakers, and a scanning device which contains driving circuits, a power supply and VHF tuner as shown in a photograph of Fig. 6. Depth of the panel is about 20 mm. The EL matrix panel is connected to the driving circuits by miniature printed-board connectors.

The circuits are all solid state. About 8,600 discrete components are mounted on printed boards. The experimental display employs integrated circuits only at a part of the counter circuit. A major portion of the circuits are currently being designed to be implemented in the IC form. However, it is difficult to substitute the IC for transistors which are directly connected to the EL elements and are supplied with relatively high voltage. Adoption of IC's would greatly reduce the number of discrete components and the size of the scanning device.

2. Performance

The experimental 52,900-element display of TV image reproduces satisfactorily static and moving images with half-tone and characters from 525-line standard TV signal in real time. Flicker and smear are not observed in the reproduced images. The cross effect is practically suppressed by the blanking pulses. However, the images it produces are still inferior to the TV pictures obtained with CRT's.

The reproduced images have highlight brightness of about 1 Foot-Lambert and contrast of 5:1 to 3:1 at normal operating conditions. It is still difficult to see the reproduced images in a lighted room. However, the reproduced images have a gray-scale range of 4 or 5 steps, with uniform brightness over the whole panel. The resolution of the reproduced images, depending mainly on the number of EL elements in the horizontal and vertical directions, is subjectively better than that of an inexpensive home video tape recorder with resolution of about 200 lines. Distortion and jitter of the images are not apparent because each EL element is addressed digitally and is provided with a light spot which does not spread out. Fig. 7 is an example of half-tone images reproduced on the 52,900-element EL matrix panel from the signal of closed-circuited TV camera. Fig. 8 is another example of a reproduced image with characters.

The life of the experimental panel is probably more than several hundred hours at normal operating condition.

Although the circuits are assembled densely, the temperature rise of the display is not so high. Power dissipation of all the display is about 100 watts.

CONCLUSION

The experimental EL display described in this paper has demonstrated the feasibility of the flat panel TV display in a diagonal scale of 13 inches. Although our EL flat panel display is aimed at the so-called wall-hanging type TV, it can be used for other display applications, such as character displays, when modified. The EL image display system has a possibliity of a color EL display, but is not readily adaptable in its present form.

ACKNOWLEDGEMENTS

The authors wish to express their thanks to Dr. S. Kisaka, Dr. Y. Iida, and Mr. Iga for their encouragements and useful discussions.

They are also indebted to Mr. S. Noma, Mr. H. Takeda, and Mr. T. Tatsumichi for their assistance to carry out the construction of the panel and the circuit.

REFERENCES

1. J. A. Rajchman, G. R. Briggs, and A. W. Lo, "Transfluxer Controlled Electroluminescent Displays", Proc. IRE, vol. 46, pp. 1808 - 1824, Nov. 1958.
2. E. A. Sack, P. N. Wolfe, and J. A. Asars, "Construction and Performance of an ELF Display System", Proc. IRE vol. 50 pp. 432 - 441, 1958.
3. M. Cooperman, "Ferroelectric Scanning of Electroluminescent Display", RCA Review, vol. 22, No. 1, pp. 195 - 205, 1961.
4. B. J. Lechner, "A Ferroelectric Transcharger Controlled Electroluminescent Matrix Display", Digest of Technical Papers, 1965 International Solid State Circuit Conference, Philadelphia, pp. 127 - 133, May 1966.
5. G. W. Taylar, "The Design and Operating Characteristics of a 1200-Element Ferroelectric Electroluminescent Display", IEEE Transactions on Electron Devices, vol. ED-16, No. 6, June 1969.
6. S. Yando, "Solid-State Display Device", Proc. IRE, vol. 50, pp. 2445 - 2451, 1962.
7. A. Vecht, N. J. Werring, and P. J. F. Smith, "High-Efficiency D. C. Electroluminescence in ZnS (Mn, Cu)", Brit. J. Appl. Phys., ser. 2, vol. 1, pp. 134 - 136, 1968.

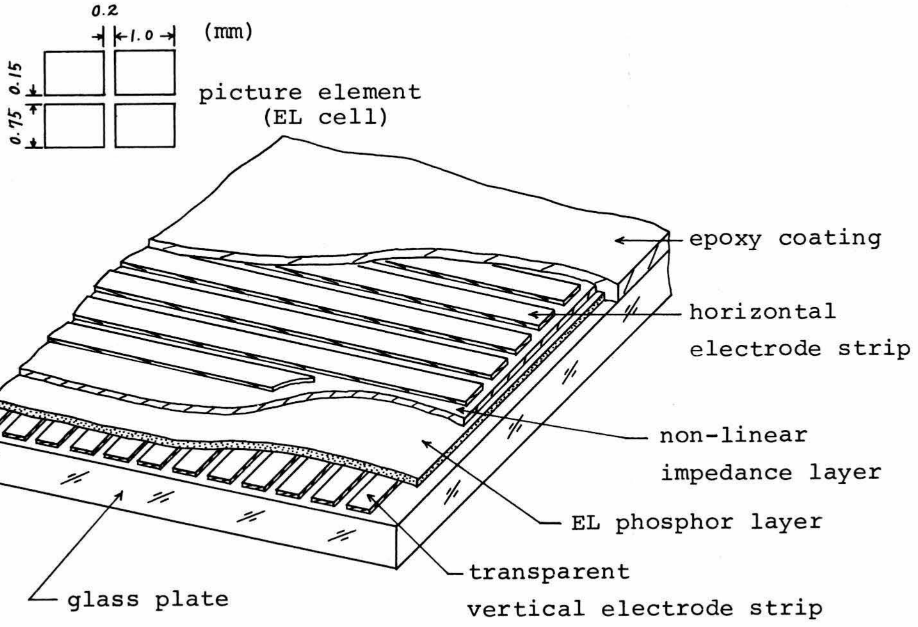

0.2
1.0
(mm)
0.15
0.75
picture element
(EL cell)
epoxy coating
horizontal
electrode strip
non-linear
impedance layer
EL phosphor layer
transparent
vertical electrode strip
glass plate

Fig. 1. Structure of EL matrix panel.

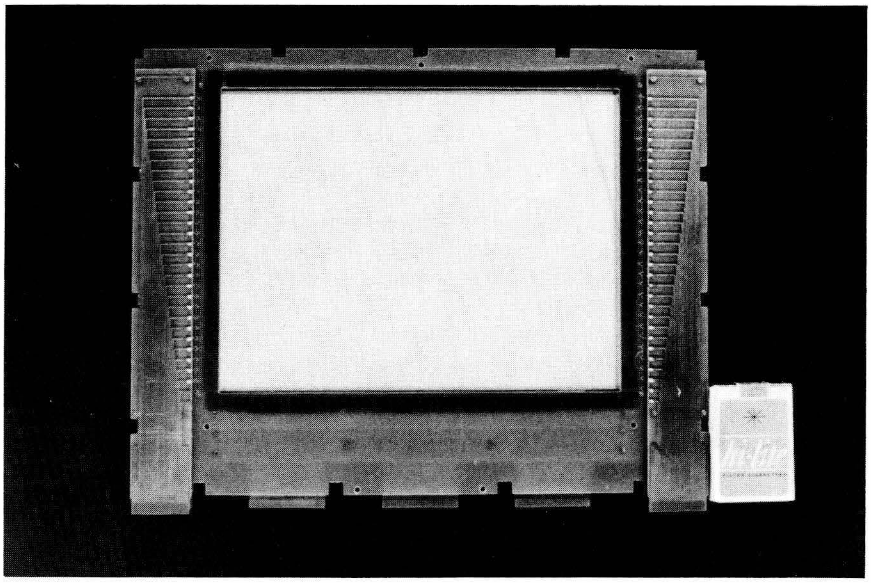


Fig. 2. Front view of 52,900-element EL matrix panel.

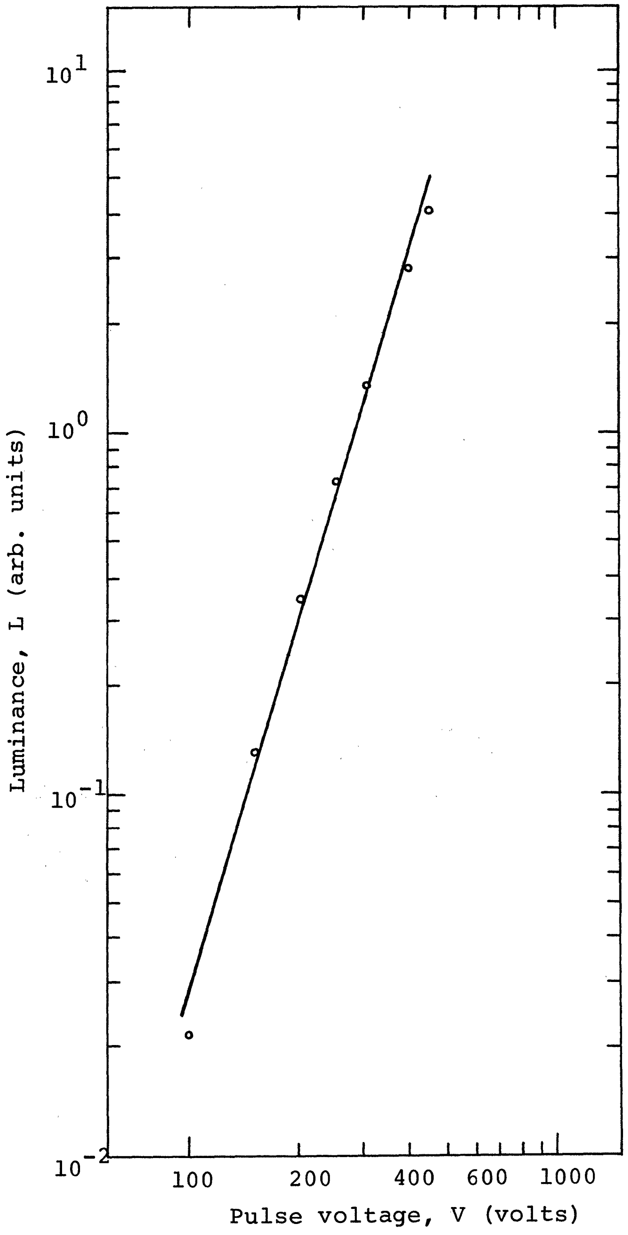


Fig. 3. Luminance-pulse voltage characteristic of
EL matrix panel.
Repetition rate, 60 Hz. Pulse width, 15 μsec.

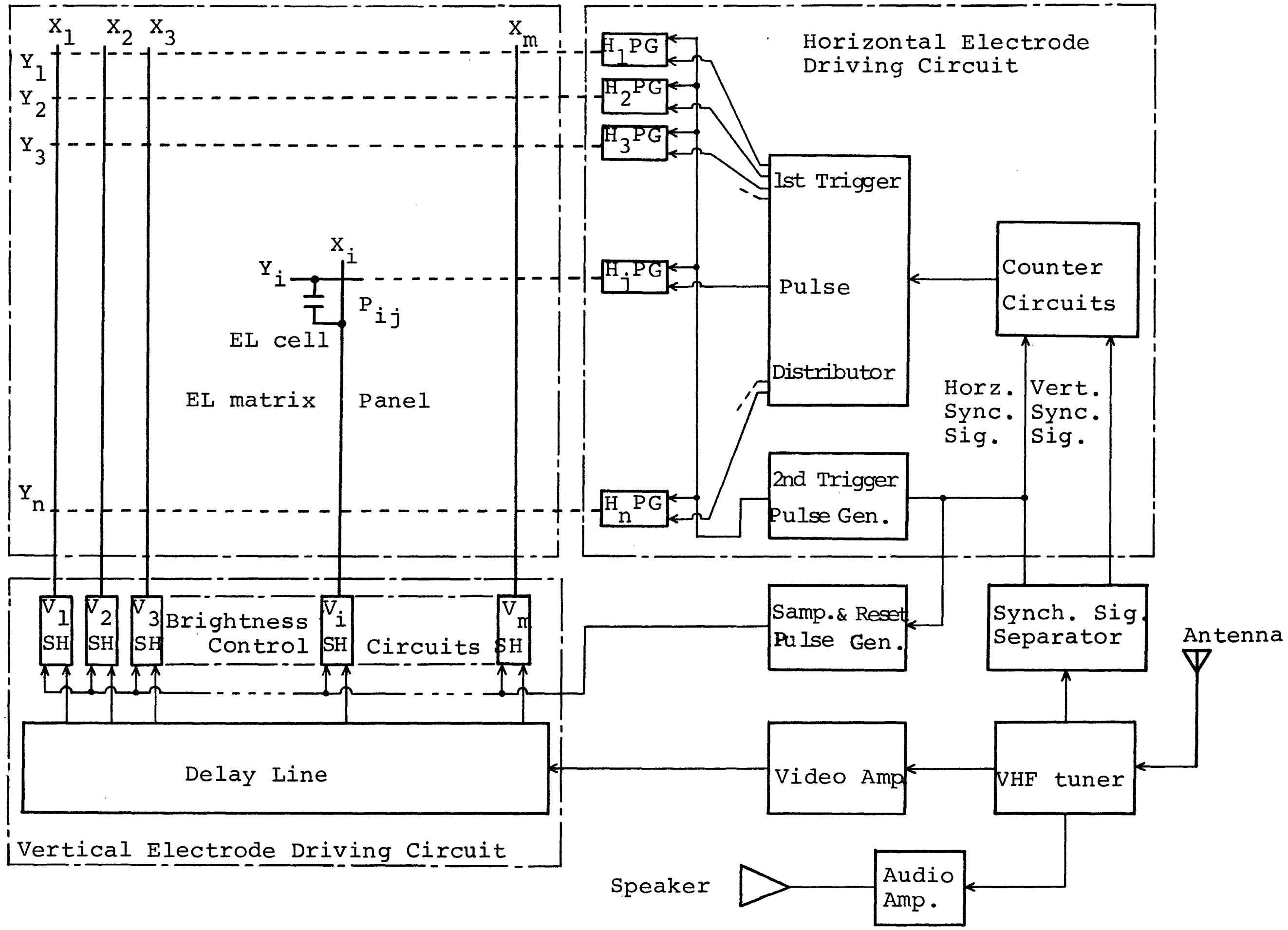
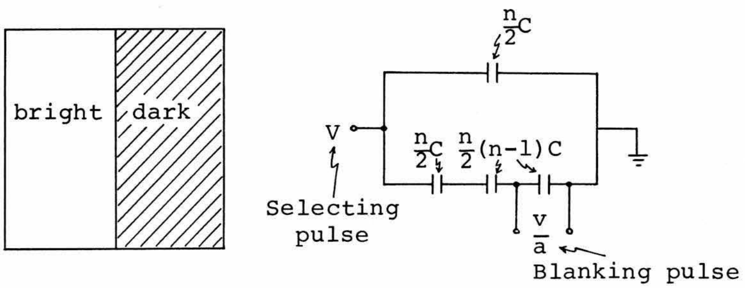

X_1 X_2 X_3 X_m
Y_1
Y_2
Y_3
Y_n
X_i
Y_i
P_{ij}
EL cell
EL matrix Panel
H_1PG
H_2PG
H_3PG
H_jPG
H_nPG
Horizontal Electrode Driving Circuit
1st Trigger Pulse Distributor
Counter Circuits
Horz. Sync. Sig.
Vert. Sync. Sig.
2nd Trigger Pulse Gen.
V_1 SH
V_2 SH
V_3 SH
V_i SH
V_m SH
Brightness Control Circuits
Samp.& Reset Pulse Gen.
Synch. Sig. Separator
Antenna
Delay Line
Video Amp
VHF tuner
Vertical Electrode Driving Circuit
Speaker
Audio Amp.

Fig. 4. Block diagram of EL image display system (TV display).


bright
dark
$\frac{n}{2}C$
V
Selecting pulse
$\frac{n}{2}C$
$\frac{n}{2}(n-1)C$
$\frac{V}{a}$
Blanking pulse

(a) (b)

Fig. 5. Diagrams for explanation of blanking pulse operation.
(a) special pattern .
(b) equivalent circuit .

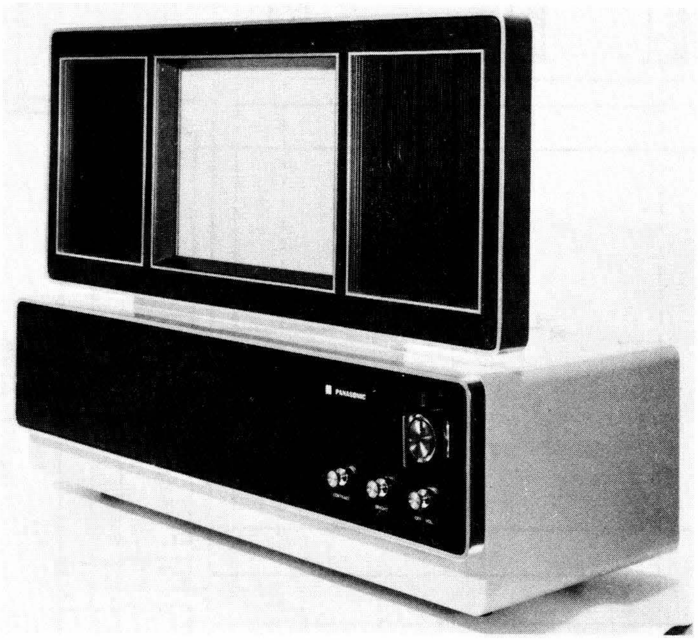


Fig. 6. Experimental 52,900-element EL display of TV image.

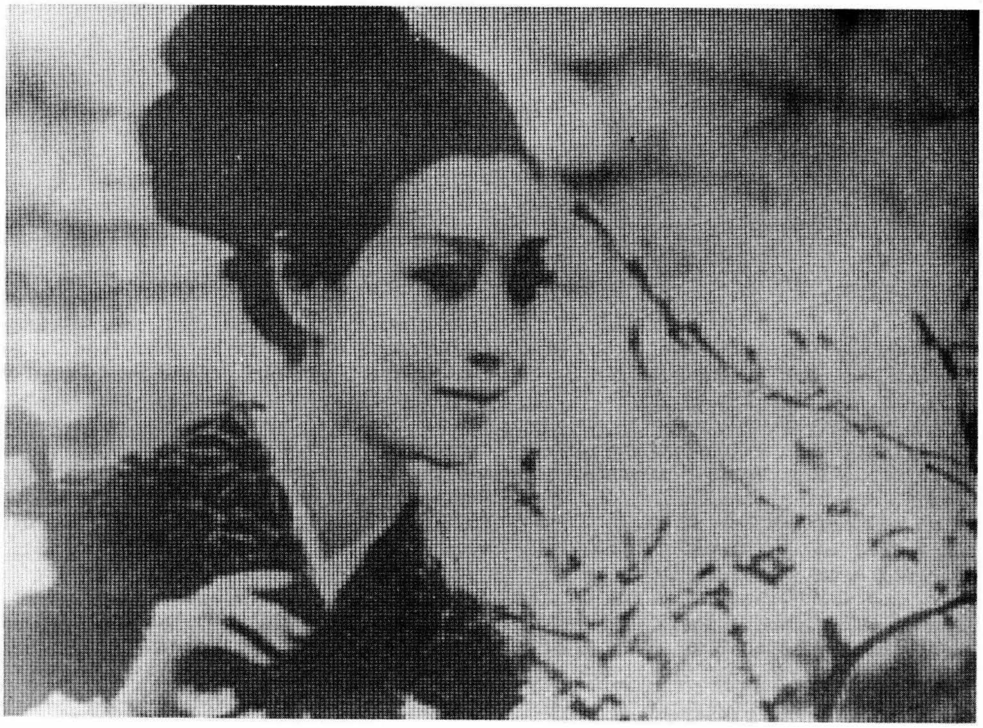


Fig. 7. Example of reproduced half-tone image.

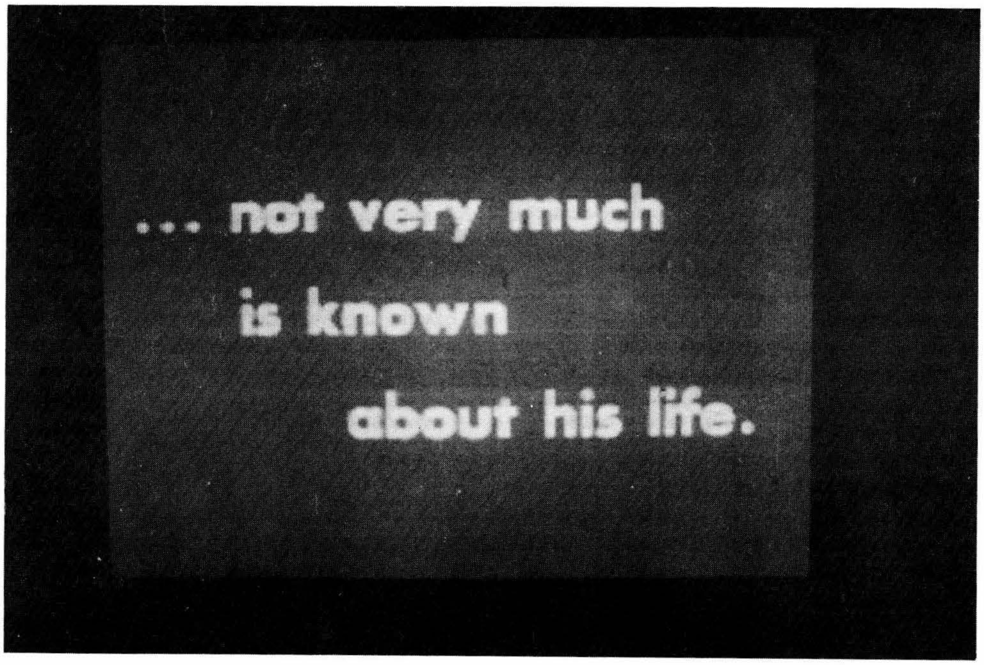

... not very much
is known
about his life.

Fig. 8. Example of reproduced image with characters .

HEAD-MOUNTED DISPLAY TERMINAL

D. L. Vickers, Computer Science, University of Utah

ABSTRACT

This paper contains a report on the present status of the head-mounted display terminal.[1] The terminal, worn like a pair of eyeglasses, contains miniature cathode ray tubes which present to the observer the illusion that he is surrounded by three-dimensional, computer-generated objects. The position and orientation of the observer's head is measured and this information is passed to high-speed digital hardware that creates a perspective view which changes as the observer moves his head. Thus, the objects should appear to be stationary as the observer walks among them.

Current hardware problems with the terminal cause observable flicker in objects defined by more than 12 lines. Only the cathode ray tube for the right eye is working, quality of the displayed lines is poor, and displayed objects don't appear stationary if the observer turns his head rapidly. Still, the illusion was reported to be realistic by over fifty persons who have used the terminal.

INTRODUCTION

One purpose for building the head-mounted display terminal was to create a special pair of eyeglasses which would allow its user (the observer) to be surrounded by computer-generated (synthetic) images (objects) which appear to have many properties of real objects. As the observer turns about, the objects enter and leave his field of view in a natural manner. As he walks around, the objects appear to be stationary; their apparent size and perspective change just as real objects appear to change. Each of the observer's eyes sees the objects from slightly different points in space to give the stereo effect we are accustomed to.

The head-mounted display measures the observer's head position and uses this information to create the proper three-dimensional perspective view of objects defined in a data file. The computing machinery from which the display was built was designed to process a data file defining 3,000 line segments and to display the resulting perspective view to the observer in less than 1/30th of a second---sufficiently fast to eliminate observable flicker.[1]

Because of problems with the matrix multiplier (a hardware device to be described later), current operating speed of the head-mounted display will facilitate flicker free displaying of only ten to twelve lines. Although far below design standards, this facility allows an observer to see simple synthetic objects which have the properties described above. Comments and reactions from over 50 persons having used the display attest to the reality of the illusion.

A feature of the display which greatly increases the reality of the illusion is that the observer can see the synthetic objects simultaneously with the real objects in the surrounding room.

Effective operation of the head-mounted display requires some rather special-purpose high-speed hardware, a streamlined software program, and some understanding and cooperation from the observer.

HARDWARE

The head-mounted display terminal consists of five separate hardware devices: a head position sensor, a general purpose digital computer, a matrix multiplier, a clipping divider, and a head set. Data is received from and/or delivered to these devices cyclically according to the order in which they are listed above and as shown in Figure 1. A summary of the functions and design criteria of the five hardware devices is given in the next few paragraphs.

Head Position Sensor

The head position sensor[1] (partially visible at the top of Figure 2) measures the position and orientation of the observer's head. A mechanical device, it is basically a shaft comprised of two concentric cylindrical tubes allowed to slide freely along their common axis. One end of the shaft is connected through a universal joint to a pivot on the ceiling. The other end is attached, also through a universal joint, to the top of the head set (described later) worn by the observer. However, the observer is not burdened by the weight of the shaft and head set because the shaft is spring loaded.

Physical limitations of the head position sensor and the surrounding features of the room in which it hangs restrict the observer's volume of head motion to about five feet square and two feet high wherein he is able to move freely,

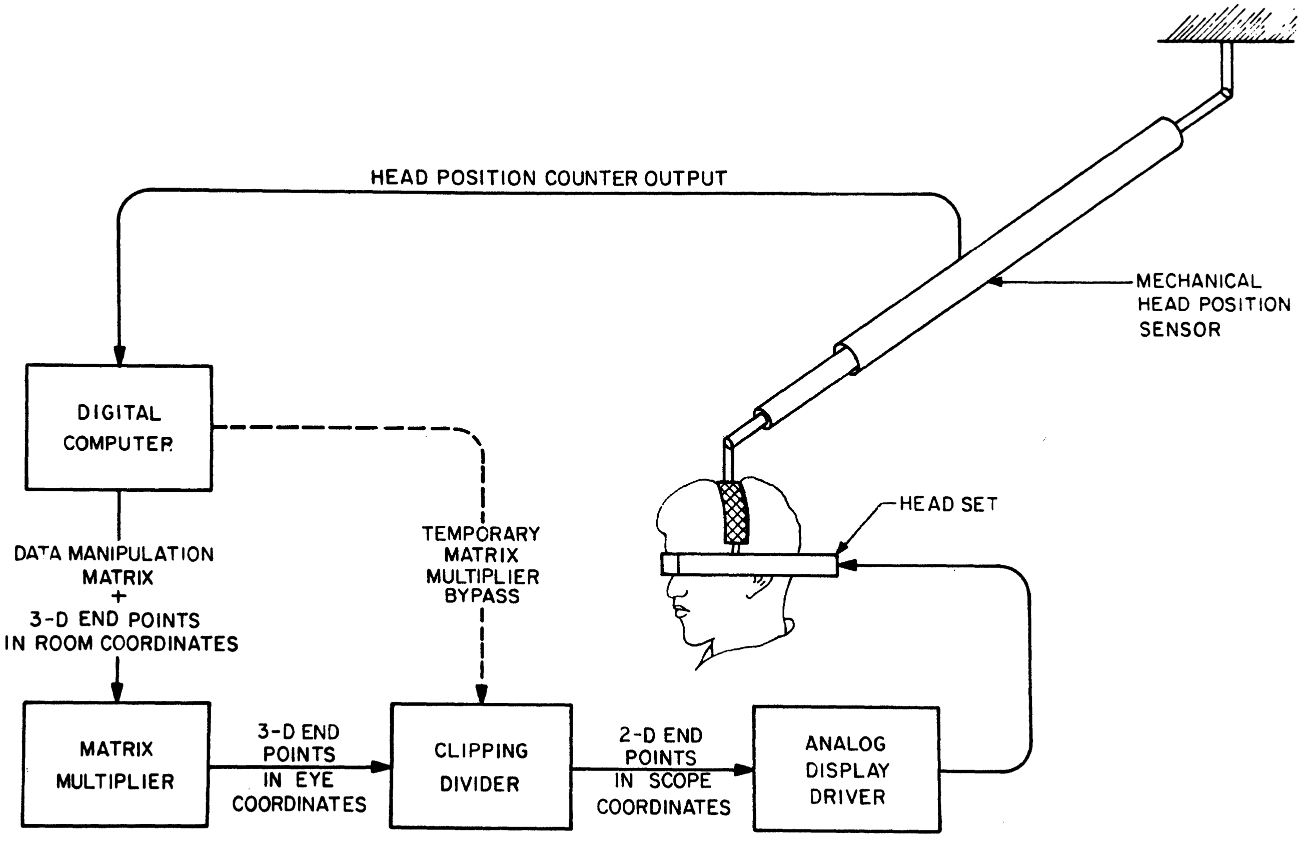

HEAD POSITION COUNTER OUTPUT
MECHANICAL HEAD POSITION SENSOR
DIGITAL COMPUTER
HEAD SET
DATA MANIPULATION MATRIX + 3-D END POINTS IN ROOM COORDINATES
TEMPORARY MATRIX MULTIPLIER BYPASS
MATRIX MULTIPLIER
3-D END POINTS IN EYE COORDINATES
CLIPPING DIVIDER
2-D END POINTS IN SCOPE COORDINATES
ANALOG DISPLAY DRIVER

Figure 1
Hardware devices and information flow in the head-mounted display terminal. The dashed line represents a temporary bypass now in use.

turn completely about, and tilt his head up approximately 33 degrees, and down approximately 80 degrees.

As the observer moves his head, six high-resolution counters measure rotation (at the universal joints near head and ceiling) and translation (along the shaft axis). These counters measure the position and orientation of the observer's head with a resolution of 1/1300 of an inch of vertical translation, 1/21 to 1/15 of an inch (depending on amount of shaft extension) of horizontal translation, and one part in 8,000 of rotation.

General Purpose Digital Computer

A Digital Equipment Corporation PDP-10 computer is programmed to interpret the signals from the six counters and create a four-by-four homogeneous matrix which defines the position of the observer's head with respect to fixed room coordinates. A new matrix is created for each frame displayed to the user.

In addition to generating this head position matrix, the PDP-10 also stores a data file which contains the coordinates of each end point describing each object displayed to the observer. A second four-by-four (object control) matrix generated in the PDP-10 defines the position of the object relative to room coordinates.*

It is this matrix which allows the illusion that the object(s) viewed by the observer as he walks about the room, may themselves be rotating, translating, and changing size independent of the observer's motion.

Multiplication of the inverse of the object control matrix with the head position matrix produces a final data manipulation matrix. Thus,

$$\begin{bmatrix}\text{Object}\\\text{Control}\\\text{Matrix}\end{bmatrix}^{-1}\begin{bmatrix}\text{Head}\\\text{Position}\\\text{Matrix}\end{bmatrix}=\begin{bmatrix}\text{Data}\\\text{Manipulation}\\\text{Matrix}\end{bmatrix}$$

---

* The object control matrix is not presently implemented. Refer to the section titled: Display File Processor.

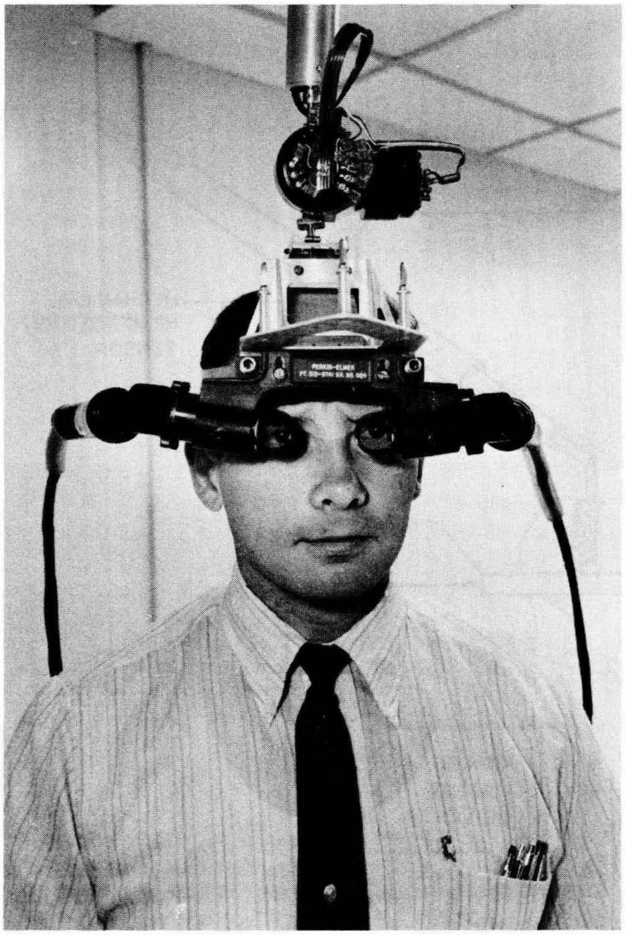


Figure 2
Observer wearing the head set of the head-mounted display.

Matrix Multiplier

The data file and data manipulation matrix created in the PDP-10 are next passed to the matrix multiplier.[1] This specially designed piece of computing hardware multiplies each end point in the data file by the data manipulation matrix and thus imposes the rotation and translation defined by this matrix onto the end point. The matrix multiplier was designed to perform the multiplication of one end point (a one-by-four matrix) with the four-by-four data manipulation matrix in five microseconds.*

Clipping Divider

Immediately following the transformation of one end point by the matrix multiplier, the end point is accepted by the clipping divider.[2] The clipping divider eliminates (clips) those objects and portions of objects behind the observer or outside his field of view and in this process computes new end points at the places where partially visible lines intersect this field of view. It also performs the division necessary to obtain a true perspective view and thus maps the three dimensional representation used in the data file into a two dimensional representation suitable for display on a flat screen.

The clipping divider is connected in pipeline fashion with the matrix multiplier on one end and the head set on the other. These connections are asynchronous to allow maximum speed in processing the data file. One design goal was that the clipping and perspective be done fast enough to process a data file of 3,000 end points in less than 1/30th of a second.

Head Set

Those end points (still in digital form) within the observer's field of view are sent in pairs to the analog circuitry which drives the head set.[1] The head set (being worn by an observer in Figure 2) presents the synthetic objects defined by the data file to the observer.

The head set has the appearance of a rather complex set of spectacles, the two temple pieces of which are miniature cathode ray tubes (CRT) each with a one half inch square screen. Optical systems in the head set magnify the pictures on each CRT and present a virtual image several feet in front of each of the user's eyes. Since each eye may be presented a slightly different view, it is possible to see a stereo image. Half-silvered mirrors in the prisms through which the observer looks allow him to see both the synthetic objects and actual furnishings and features of the surrounding room simultaneously. Thus, material displayed by the head set can be made to hang disembodied in space or to coincide with maps, desk tops, walls, or the keys of a typewriter.

SOFTWARE

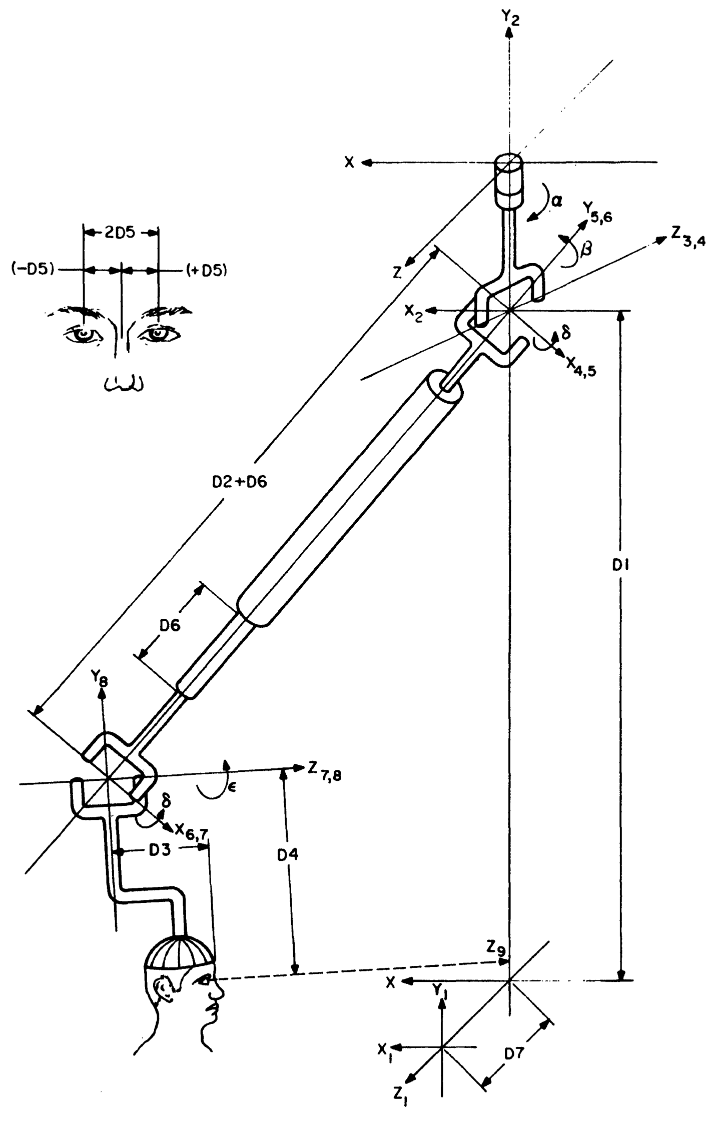
All the software programming for the head-mounted display is done in MACRO 10 assembly language on the PDP-10. The program currently consists of two parts: a head position sensor processor, and a display file processor. Computer control is passed from one to the other as each completes its task.

Head Position Sensor Processor

The head position sensor processor reads the six head position sensor counters, converts the counter values to distances or angles as required, and forms the necessary four-by-four rotation and translation matrices (MHDPS1-8 in Figure 3). These matrices along with ELBMX1-2, which allow for an initial 15 degree (E) bend in the lower universal of the head position sensor, are then multiplied together to form the head position matrix. This matrix transforms the data file from room coordinates to eye coordinates (a coordinate system attached to the eye of the observer). Ultimately, a head position matrix will be computed for each eye; currently, however, only the right CRT of the head set is working and there-

---

* The matrix multiplier is currently inoperable. Refer to the section about CURRENT SYSTEM STATUS.



$$
\overset{\text{DATA FILE}}{\begin{bmatrix} X_A & Y_A & Z_A & 1 \\ X_B & Y_B & Z_B & 1 \\ X_C & Y_C & Z_C & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}_1}
\overset{\text{MHDPS1}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -D1 & D7 & 1 \end{bmatrix}_2}
\overset{\text{MHDPS2}}{\begin{bmatrix} C\alpha & 0 & S\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -S\alpha & 0 & C\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_3}
\overset{\text{MHDPS3}}{\begin{bmatrix} C\beta & -S\beta & 0 & 0 \\ S\beta & C\beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_4}
\overset{\text{MHDPS4}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\gamma & -S\gamma & 0 \\ 0 & S\gamma & C\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_5}
\overset{\text{MHDPS5}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & (D2+D6) & 0 & 1 \end{bmatrix}_6}
\text{-----}
$$

$$
\overset{\text{MHDPS6}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\delta & -S\delta & 0 \\ 0 & S\delta & C\delta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_7}
\overset{\text{ELBMX1}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & CE & SE & 0 \\ 0 & -SE & CE & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}
\overset{\text{MHDPS7}}{\begin{bmatrix} C\epsilon & -S\epsilon & 0 & 0 \\ S\epsilon & C\epsilon & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_8}
\overset{\text{ELBMX2}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & CE & -SE & 0 \\ 0 & SE & CE & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}
\overset{\text{MHDPS8}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \pm D5 & D4 & -D3 & 1 \end{bmatrix}_9}
=
\overset{\substack{\text{TRANSFORMED} \\ \text{DATA FILE}}}{\begin{bmatrix} X'_A & Y'_A & Z'_A & 1 \\ X'_B & Y'_B & Z'_B & 1 \\ X'_C & Y'_C & Z'_C & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}_9}
$$

Figure 3
Transformation of data file through the head position sensor from room coordinates (subscript: 1) to eye coordinates (subscript: 9). SX and CX represent sine and cosine of X.

fore the head position matrix transforms just to the right eye.

Display File Processor

The display file processor walks through the data file converting (when necessary) the end point descriptions to the proper format, scaling the end points, and possibly adding an initial x, y, or z offset to each end point. This processor then presents each end point to the matrix multiplier* which previously has been loaded with the proper data manipulation matrix.

Right now, the data manipulation matrix is identical to the head position matrix but future plans include pre-multiplying the head position matrix by the inverse of an object control matrix which will allow both dynamic change in the position, orientation and size of objects, and dynamic change (creation or deletion) of the object descriptions. Thus, an observer using the head-mounted display may be able to walk within a simulated model of our solar system and while watching it spin he could create a comet of some known mass and velocity and see from any vantage point he chooses, the mutual effects of the comet and planets on the other's orbit. Currently, without the implementation of the object control matrix, objects can be scaled and translated relative to room coordinates, but must appear stationary to the observer as he walks about.

After the display file processor has sent the last end point of the last object named in an object list to the matrix multiplier, PDP-10 control is passed back to the head position sensor processor.

CURRENT SYSTEM STATUS

The previous section about software alluded to several examples where both hardware and software presently fall below design expectations or desired levels of performance. One example was the mechanical failure of the CRT for the left eye; another was the exclusion of an object control matrix from the software facilities. Before describing what an observer can see and how he reacts, it will be necessary to describe the current performance level of the head-mounted display terminal.

The head position sensor is working well and as described. An ultrasonic version of the head position sensor, using transmitters mounted on the head set and receivers mounted in the room, has been constructed but the software to interpret its output is not yet perfected.

Some of the software in the PDP-10 was done in haste and needs a great deal of streamlining. The software package is not yet capable of feeding the end points within the data file to the matrix multiplier at the rate the multiplier was designed to accept them.

This software problem, however, is not critical since the matrix multiplier is not currently working.** Instead, the functions of the matrix multiplier are simulated in software and the PDP-10 passes data directly to the clipping divider.

Although running at only 5/8 of its design speed, the clipping divider is not limiting the refresh rate of the display system. It does, however, occasionally give erroneous data to the display-driving analog circuitry and thereby causes spurious lines or "glitches" to appear on the head set.

In addition to the problem with the left CRT, the head set has some trouble in its display-driving circuitry. The quality of the lines displayed is poor in that only about 7/8 of each line is drawn, leaving space between line segments which should be contiguous. Too, short lines are brighter than longer ones.

Regardless of these numerous maladies, the head-mounted display does work and an observer can see objects suspended in space and can walk around and view them from different angles.

USING THE DISPLAY

Before an observer is free to walk about looking at synthetic objects, the head set must be fastened securely to his head and adjusted to his individual features. For fastening, the head set is equipped with an adjustable harness that fits like a cap and with plastic temple pieces similar to those used in eyeglasses. Automatically-adjusting forehead pads help fix the distance between the observer's eyes and the optical system.

Both mechanical screw adjustments and software programming adjustments can be used to allow for variations in the distance between the eyes of each observer. To insure that the adjustments are accurate, a computer-generated pattern of spots can be made to coincide with a similar pattern on a wall or walls of the room. These adjustments, however, are usually not necessary for satisfactory performance of the display system.

In order to see the displayed objects, it is necessary that the observer's head does not shift appreciably within the head set. Because of the head position sensor and the weight of the cables going to the cathode ray tubes, a sudden movement of the observer's head may cause such a shift. For this reason, most observers prefer to further secure the head set to their head by holding on to the cathode ray tubes with their hands. One observer suggested adding a chin strap to the harness. An obvious improvement regarding head shifting would be substituting the ultrasonic head position sensor for the mechanical one now in use.

---

* Refer to CURRENT SYSTEM STATUS.

** Although it was working at Harvard University, where the head-mounted display was designed and constructed by Ivan E. Sutherland as principal investigator, the matrix multiplier is temporarily inoperable due to its journey to the University of Utah and to several subsequent relocations. Actually, the only serious problem with the matrix multiplier is that its maintenance lies in the hands of some very much over-worked graduate students.

Cubes

With the head set finally adjusted, the observer's attention quickly turns to the objects he sees. The standard demonstration depicts two cubes, one centered within the observer's head as he stands directly under the head position sensor mounting (Figure 4). The other cube is "fastened" to a wall four feet away. Both cubes are six inches on a side and are at eye level for most observers. They are "wire frame" drawings and no attempt is made to blank lines that would be hidden were the cubes solid.

Figure 4 was made by superimposing a cube photographed from a standard oscilloscope on a picture of the room which houses the head-mounted display. The composite picture is very similar to the view an observer sees while wearing the head set. The oscilloscope from which the cube was photographed even draws lines of incorrect length as does the head set (though the head set lines are somewhat worse). Also, a clipping divider glitch can be seen extending from the lower right corner of the cube.

An interesting fact about the head-mounted display is that with no movement of his head, an observer cannot say whether the cubes are large and far away or small and near. Yet most observers when requested to do so could estimate the size and position of the cubes quite accurately by moving their heads only a couple of inches. One trait common to those having difficulty with the estimations was to overestimate the size of the cube nearest them. Possibly the problem here was that people simply aren't used to looking at a six inch cube from a distance of two or three inches.

After walking around the cubes and aligning his eye first with one side and then the other of each cube, an observer can tell exactly the position, orientation and size of the cube.

Quality of the Illusion

Bypassing the matrix multiplier slowed the refresh rate sufficiently to cause an observable flicker when the two cubes are displayed. One may think that this flicker, coupled with the clipping divider glitches and the 7/8 length lines, would destroy the effectiveness of the illusion. This was not the case, however. Apparently the ability to move around and view the cubes from many angles and distances is the prime factor contributing to a realistic illusion. Some observers weren't even aware that the illusion was visible only to the right eye. Another factor which tends to make the illusion more realistic is the background lighting in the room.

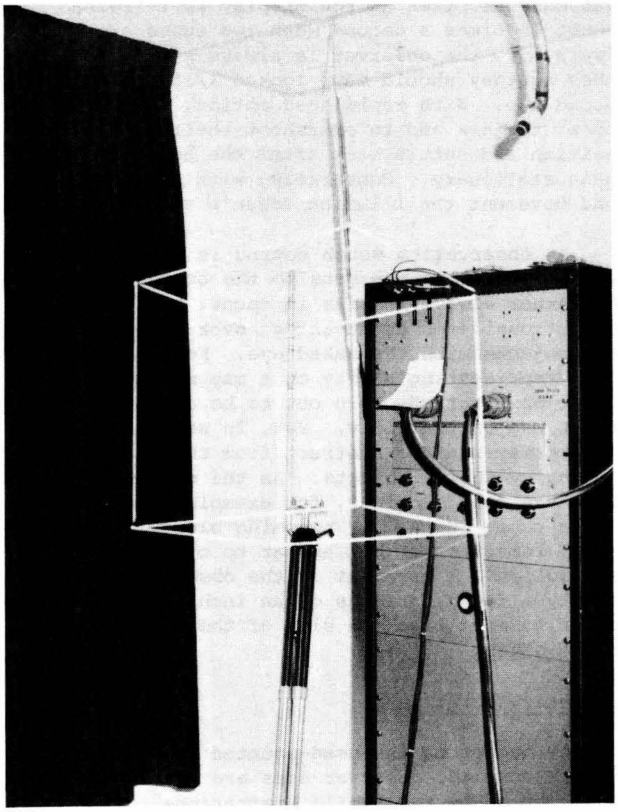


Figure 4
Cube and surrounding room photographically simulated to appear as seen through the head-mounted display.

It was found that in a lighted room the cubes appear more stationary probably because the observer can then compare their positions to features in the room; also, the flickering is less noticeable with a lighted background. Even with the room lights at full intensity, the cubes are visible against dark surfaces and shaded walls.

One observer commented that the cubes seemed more natural when he didn't hold on to the cathode ray tubes with his hands. With his hands in his pockets he felt the head set seemed more like any other pair of eyeglasses.

Several factors tend to distract from the reality of the illusion. One such factor is the limited field of view of the optical system. Although the observer can see the room around him with his normal field of view, his field of view of synthetic objects is limited to less than 40 degrees. This is particularly noticeable when one cube is enlarged to six feet on a side. The observer standing inside the cube and looking out horizontally can see parts of the floor and ceiling of the room, but can see only the middle portion of the cube edges.

Another distraction occurs when the observer turns his head rapidly from side to side. Since the head position sensor counters are read only as often as the display is refreshed (about 15 times a second when two cubes are displayed), the observer is always seeing the cubes as they should have looked 1/15th of a second ago. With rapid head motion, the cubes appear to jerk and to overshoot their real position and settle back after the head is again stationary. Conversely, with slow, steady head movement the illusion doesn't suffer.

An observation worth noting is that regardless of how close one gets to the corner of a cube, the corner always remains in focus. This quality can be considered an advantage over looking at real objects with the naked eye. For instance a "dot" representing a city on a map may, upon much closer observation, turn out to be a detailed street map of that city. Yet, in some cases, this quality may tend to subtract from the realism in viewing synthetic objects. As the observer's eye nears the cube's corner, for example, the lines of the cube, instead of becoming blurry and out of focus, remain clear but appear to overreact to head motion. A movement of the observer's head of only a few hundredths of an inch causes the corner to move from one side of the field of view to the other.

Parameter Variation

One aspect of the head-mounted display is that objects the observer sees are not necessarily bound by real-world limitations. For example, one could write a program to interpret the observer's head motion to be approaching the speed of light. Then as the observer walked forward he might see some extreme effects of length contraction on the synthetic objects.

Although the speed of light experiment has not yet been performed, one observer, by doubling the effective counter value for the counter which measures rotation about the vertical axis, noticed that as he walked around a cube, the cube appeared to revolve around this vertical axis. Another experiment was tried where all the counter values were complemented. This gave the sensation of "chasing rainbows". As the observer approached a cube, it got smaller; by turning his head to the right, the cube moved to the right of his field of vision.

United States Map

Data files for objects other than cubes have been tested. The United States map shown in Figure 5, for example, is made up of over 700 line segments. If the matrix multiplier were working, it would be possible, theoretically, to see this map flicker free as it "hung" on a wall of the room. Under the circumstances, the map hangs on the wall, but it takes two full seconds for the display to cycle through the data. A flicker rate of two seconds is intolerable when the observer moves his head at all. Even when the head is still, a flicker rate that slow is very disturbing to look at.

The map does show some problems with line quality. Although the picture shown here does not portray the three-dimensional effect, the head position sensor, three-dimensional software, and clipping divider were all being utilized. This photograph, too, was made from a standard oscilloscope. The problems of incorrect line length and more intensity on shorter lines are evident in Figure 5. Still, it is clearly recognizable as a map of the United States of America.

General Impressions

Miscellaneous comments and reactions from observers were quite varied. One said he got eye strain from the display; another complained of a headache from the head straps. Several didn't like the restrictions in head motion caused by the head position sensor.

Those wearing eyeglasses (author included) were faced with the choice of removing the glasses and squinting at blurred objects (as the observer in Figure 2 appears to be doing), or leaving the glasses on and not being able to take full advantage of the head set straps and forehead pads. It might be wise to add a variable lens system between the head set prisms and the user's eyes to allow for individual adjustment.

Observers using the head-mounted display numerous times seemed to become less and less conscious of the weaknesses of the display.

SUMMARY

Even with a noticeable flicker, improperly drawn and unevenly illuminated lines, frequent glitches, jerking and overshooting, and the left CRT gone, the illusion of being surrounded by three-dimensional synthetic objects appeared to be real. Most observers indicated they were awed or at least impressed by the reality of

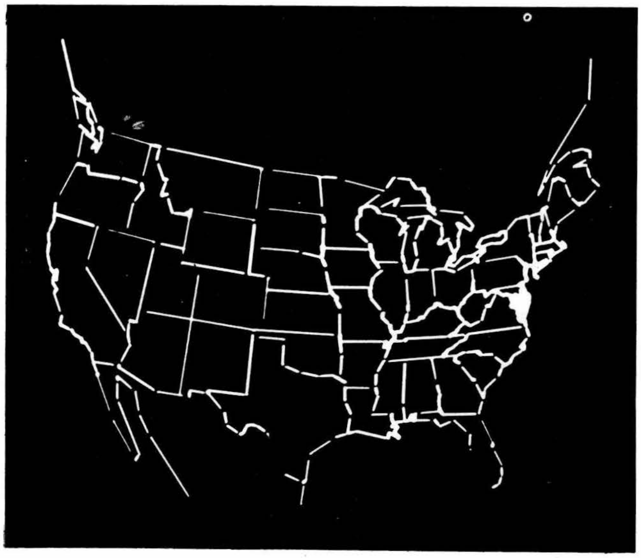


Figure 5
Map of the United States produced by the head-mounted display hardware.

this illusion. All were able to respond correctly to questions aimed at checking this reality. Such questions were: "What objects do you see?"; "How big are the cubes?"; "How far away are they?"; "Do they appear to be stationary or to move as you walk around them?"

These results lead one to believe that the observer's own motion contributes a great deal to the illusion's reality and that observer's tend to be quite tolerant of many imperfections in the head-mounted display terminal.

ACKNOWLEDGMENTS

The author thanks Ivan E. Sutherland for his cooperation and direction in the preparation of this paper, and Jim Callan of Evans and Sutherland Computer Corporation for the United States map data. The Evans and Sutherland Computer Corporation has been very helpful in supplying information and data.

This research was sponsored by the Advanced Research Projects Agency, Department of Defense, and was monitored by AFSC, Research and Technology Division, Rome Air Development Center, Griffiss Air Force Base, New York 13440, under Contract AF30(602)-4277.

REFERENCES

1. Sutherland, Ivan E., "A Head-Mounted Three-Dimensional Display", Proceedings of the Fall Joint Computer Conference, 757-764 (1968).

2. Sproull, R. F., and Sutherland, Ivan E., "Clipping Divider", Proceedings of the Fall Joint Computer Conference, 765-776 (1968).

EYE TRACKING AS A COMPUTER INPUT TECHNIQUE

Sam L. Geffner, Project MAC, Artificial Intelligence Group, M.I.T.

ABSTRACT

One kind of computer input device which has great potential is the oculometer, an instrument which measures point of regard of the computer console user (point at which he looks). In conjunction with a computer generated display which can then change in real time as a function of point of regard of the subject, the oculometer makes possible a variety of interactive display consoles. Practical applications of such schemes have had to await the development of a design which doesn't inconvenience the subject, and which conveniently interfaces with the computer. Such an oculometer design is described, making use of an infrared light source reflected specularly off the front surface of the cornea and the diffusely off the retina through the pupil. The relative motions off these two reflections give an eye angle measurement independent of head angle, permitting the console user reasonable freedom of motion. Application areas currently under exploration include: using the eye as a control device; recording fixation and exploration patterns; game playing; training machines; and psychological testing and recording. These application areas are depicted by a number of examples, some speculative and some which have already been accomplished.

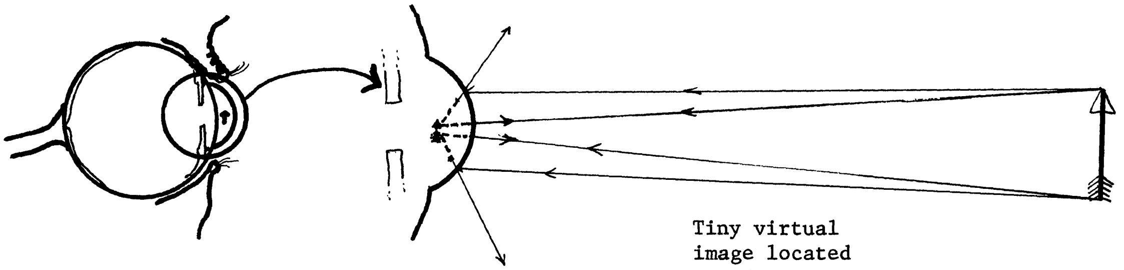
I would like to describe a device that is likely to have a number of uses in the future as an input device for computer terminals. This device is called an oculometer. It measures properties of the human eye in such a way as to determine where the eye is looking (point of regard). (This information can then be fed in real time to a computer.) I want to go on to describe the unique properties of this oculometer, which make it a potentially feasible input device, and I will describe some applications in which it has a clear superiority over other input devices.

For me, the original motivation for an eye tracking device sprang from psychological investigations of Gilbert Voyat, now at Yeshiva University. He wanted to determine the role of eye motions in visual perception: the extent to which it is necessary for there to be eye motions, causing the observed image to slide about the retina, for certain kinds of perception to occur. Now some visual data can be absorbed tachistoscopically where the very brief flash presentation prevents significant eye motion. In other cases, eye motion is absolutely necessary for perception. We wished to investigate the role of eye movements for a certain class of optical illusions. The conventional techniques for performing such experiments were tedious; the available eye tracking techniques were all relatively inconvenient. Let me give a few examples of conventional techniques. The movingmirror contact lens system affords high accuracy but required that a contact lens be ground to fit each experimental subject. A mirror fixed to the contact lens would reflect a light beam onto a photocell receiving system which could then caulculate eye angle. The subject's head had to be rigidly fixed. Another technique is electro-oculography, in which amplified corneo-retinal potentials give a measure of eye angle. With the head fixed, the eye angle record could be transformed into a point-of-regard record. (Point-of-regard is the point whose image falls on the fovea of the eye.) This technique required electrical connections to the head. Or one could take a photographic record of eye-movements. I could go on to list a number of other techniques, but all tended to have disadvantages either in the form of subject discomfort, or difficult subject preparation, or both. All methods left one with a record to be analyzed either by hand, or laboriously encoded for digital analysis.

What was needed was an eyetracking technique which conveniently interfaced with a digital computer, did not require any special subject preparation, and did not inconvenience the subject. With such a device an experiment could be performed with a large number of subjects, with a variety of stimuli presented by the computer, and with the eye movement response automatically recorded and statistically analyzed. This application, which is of very special interest, provided the initial motivation for the development of a technique which promises to have a rather broad application as a computer terminal; again because of a) convenient real-time interface with computer b) no subject preparation and c) minimal inconvenience.

Let's consider the structure of the eye for a moment. In a cross section the corneal bulge protrudes from the sphere of the eyeball. As you can observe by looking in an eye, this highly reflective, highly convex corneal surface reflects a size-reduced virtual image of any bright object in the field of vision.

FIGURE 1: CROSS SECTION OF EYE: CORNEAL REFLECTIONS



Tiny virtual image located inside the cornea

The image appears to be within the cornea itself, and will move as the eye moves. Many eyetracking techniques have attempted to use the motions of this image as a measure of eye pointing angle. In fact, the image of a point light source will deflect from side to side in a way nearly proportional to eye angle. However, this motion is extremely small relative to the motions of the head, and so the head must be fixed by bite bar. This is a difficulty common to most eyetracking techniques: because the movements of the eye are equated to changes in eye angle, a reference measurement is needed to cancel out head motions. In January 1968, in work done for my master's thesis, I described a system which used the iris-sclera boundary (the sclera is the "white" of the eye) as a reference, thereby obtaining a measure of eye angle independent of head motion. The system worked by viewing the subject's eye with a vidissector which was driven by a PDP-6 computer to track the image boundaries. This primitive system gave about 1/2 degree of accuracy in the horizontal axis. But the pupil iris boundary had relatively low contrast and the facial features such as eyelashes, etc. contributed to noise and unreliability. However, this early work led us to consider an instrument using a modified scheme developed by Honeywell for NASA and under NASA contract we have implemented computer tracking programs to drive this instrument. These digital computer programs replace the original analog computer driving circuitry, which had restricted the usefulness and reliability of the instrument. The digitally-driven version of the oculometer was good during slow eye movements to 1/2 degree of accuracy in both the horizontal and vertical axes as this paper went to print. We are now developing programs to measure the oculometer's accuracy with higher precision, especially during rapid eye movements, when accuracy may be rather worse. Reliability is excellent, semi-automatic calibration features are provided, and imminent improvements are expected to increase accuracy by a factor of two.

The NASA oculometer operates in the following ingenious way:

FIGURE 2: THE OPTICAL SYSTEM: PUPIL IMAGE RAYS

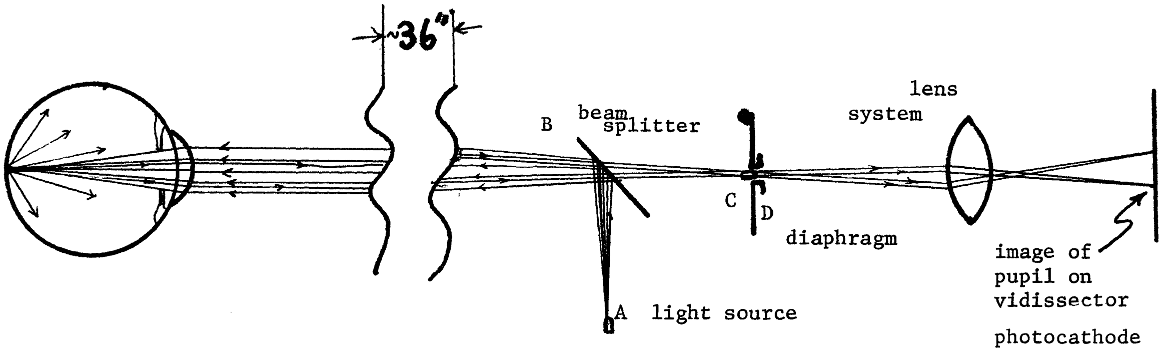
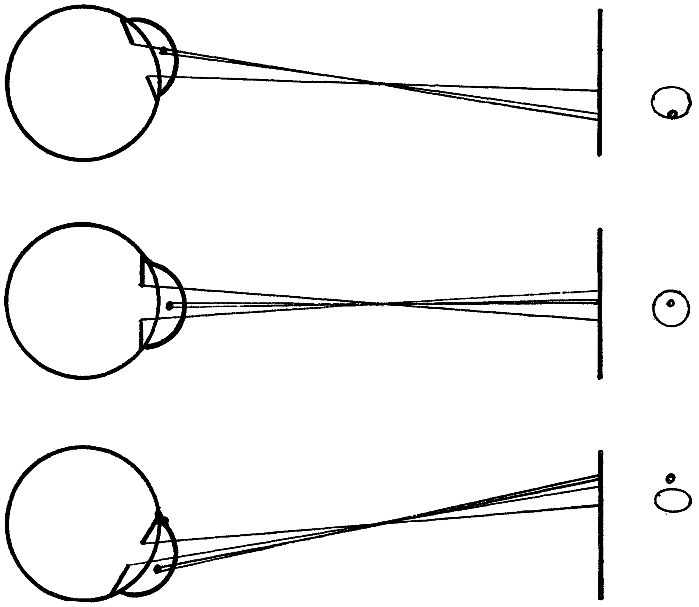


image of pupil on vidissector photocathode

A small diameter collimated light source at A is reflected off a beamsplitter mirror at B towards a subject's eye. Those rays of light reflected off the corneal surface form a tiny virtual image of the source whose apparent location is just inside the cornea. (These rays are not shown.) Other rays from the source enter the eye, forming a more or less sharply focused image at the back of the retina. Since the retina is a diffuse reflector, some of the light is scattered back toward the pupil, and returns exactly along its former path to form a real image of the light source A just at the light source, as well as another one at C as a result of rays passing through the beamsplitter. These rays at C (both corneal and retinal) pass through a small diaphragm D whose diameter is about that of the light source A whence a conventional lens assembly images the corneal and retinal images onto the vidissector photocathode. The eye appears to the vidissector as a bright disc (the pupil, with light returning through it) with a brighter dot inside (the corneal reflection). These two images are located at different distances from the eyeball's center. (Figure 3.) Thus, as the eye rotates, these image features will move relative to each other, giving a measure of eye angle independent of head position. The small diameter of diaphragm D results in very high contrast between the lighted eye and the rest of the face without reduction in signal since the returned light from the pupil is all focused through that small area. The use of an infrared light source permits a minimal disturbance of the subject's normal visual patterns. Because of the collimated light source and optical geometry this technique is insensitive to minor head motions, and it may even be feasible to track and record the eye motions of animals.

FIGURE 3: RELATIVE MOTIONS OF CORNEAL AND PUPIL IMAGES WITH EYE ROTATION



We have filmed a variety of experiments, some conducted early in 1969 when the NASA instrument was briefly on loan and some in the last year during our NASA contract period. These explore some applications of an oculometer-computer configuration, using the ability of the computer to generate images in real time, interactively responding to eye motions measured by the oculometer.

FILM:

- THE SETUP
- ITS USES
  - AS A WRITING INSTRUMENT
  - AS A SCANNING DEVICE
    - DESIGN APPLICATIONS
    - FOR TESTING-DIAGNOSIS
  - AS A CONTROL DEVICE
    - FOR GAMES
    - FOR AIRTRAFFIC CONTROL
  - AS A TRAINING DEVICE
    - READING
  - AS AN INFORMATION RETRIEVAL DEVICE
    - MAP READING

What are the situations which eye tracking is the appropriate mode of input? In particular, in what future instances might an oculometer be an important part of a computer console? While it's hard to give a general answer, there are certainly a variety of special applications in which eyetracking now looks like the best way.

1) Psychological applications: We have already described how the real time interactive connection with computers gives ease and flexibility in running psychological experiments related to visual perception.

2) Alertness monitoring, etc.: Since eyetracking takes no special effort on the part of the console user, this information is available nearly free. Visual interest is positively correlated with changes in pupil size, and the size and frequency of saccidic motions currently seems to be related to attention. Blink rate is correlated with stress or fatigue. Such "body language" seems to give a truthful, automatic, and rather hard to disguise emotional feedback from the console user. This may be of increasing importance as consoles become more sophisticated.

3) As an input device for visually controlled processes eyetracking has great potential. The map reading experiment is an example of such a system working in a way that depends on eye tracking to achieve high efficiency. In other visual control processes, eye feedback may or may not be optimal. In a series of visually controlled oculometer-versus joystick-controlled tracking tests, a range of moderate speeds emerged in which the eye was superior to the joystick. At low speeds, the steadiness of the joystick was an advantage, but it's clear that, at least sometimes, even though contrary to tradition, the eye is quicker than the hand.

4) As a training device; in the learning of all types of complex processes an orderly direction of attention to various aspects of the process is essential: in reading, the ordered left to right, top to bottom motions of words scanning; in instrument panel flying, a less structured kind of looking which nevertheless never neglects any of a large number of instruments. In teaching machines, eye motion monitoring could cause the simulator to respond with a problem which could positively or negatively reinforce visual behavior. In a system to train certain kinds of reading patterns, a computer system could, for example, force an onward eye movement pattern by displaying only words still to be read. Once scanned, a word would disappear. In an instrument flight simulator, for example, if the pilot failed to examine an altimeter for a long period of time, a simulated altitude loss ultimately resulting in a crash (simulated, we hope) might be an appropriate way of negatively reinforcing this maladaptive behavior. Such systems might also be of use in training X-ray readers or reconnaissance photograph readers.

The above suggested applications merely point in the direction of a variety of broad areas. Since convenient, computer interfaced eyetracking has only recently been available it's difficult to predict with certainty in which of the above mentioned application areas eyetracking may come into general use. However, it seems probable that in at least some instances, an eyetracking device, perhaps of the kind I've described, will be part of consoles of the future.

The research reported in this paper was sponsored by the Electronics Research Center under NASA Contract 12-2093.

REFERENCES

Geffner, S.L., "An Eye Tracking Program", Master's Thesis, M.I.T., (1963)

Merchant, J., "Interim Report: Laboratory Oculometer", Honeywell Radiation Center, Lexington, Mass. (1968)

Merchant, J., "Interim Report: Design of the Advanced Remote Oculometer", Honeywell Radiation Center, Lexington, Mass. (1969)

SPEECH INPUT TERMINALS FOR COMPUTERS: PROBLEMS AND PROSPECTS *

D. R. Reddy, Computer Science Department, Carnegie-Mellon University

ABSTRACT

Scientific and utilitarian aspects governing the cost, usability and availability of speech input terminals for computers within the next decade are considered. The structure of a real time speech recognition system presently working in a time-sharing environment is described to illustrate the differences in approach between classical pattern recognition and speech recognition. Several unsolved scientific and engineering problems are presented which must be solved at least partially before we can expect to have speech terminals for computers.

INTRODUCTION

It is not surprising that many scientists and engineers should have thought about building machines that could understand speech. What is surprising is how many of them grossly underestimated the effort required to build a non-trivial speech recognition system. In spite of many years of research, not only are we not in a position to provide speech input terminals for computers but cannot even answer satisfactorily a few essential questions about them. What type of a speech input terminal can we expect to have within the next decade? Will it be usable, i.e., reliable, accurate, fast, etc.? Who would want to use it? What would it cost? The answer to these, and other similar questions, is 'we are not sure'. What is more, there is no clear plan at present to obtain reliable answers to these questions. All that can be done is to see what has been accomplished and what problems remain to be solved before we can begin to answer these questions. To this end, this paper presents an evaluation of the state of the art, describes the structure of a real time speech recognition system presently working in a time-sharing environment, and discusses several unsolved problems which must be solved, at least partially, before we can expect to have speech input terminals for computers.

Why do we need speech terminals? We seem to be doing fairly well with cards, tapes, keyboards, and CRTs (Pierce, 1969). Why waste our resources on this area, especially when it looks like no speech input terminal we can hope to build in the foreseeable future is likely to converse in spoken English with the facility of a native speaker in a noisy environment? Such a comarison would not be entirely relevant since we don't now use English, or noisy telephone lines, for man computer communication with the existing I-O devices.

A more appropriate question would be whether there exist situations where a speech input terminal is needed and where the presently available devices are not satisfactory.[1] There are several simple tasks which are worthwhile and can be done using a limited vocabulary word recognizer. The main problem here is that the people who are not in favor of speech recognition research claim, and rightfully so, that "anything" you can do with a limited vocabulary recognizer you can do with a specailly designed box of function keys.

Clearly, if we had a computer system which can do half as decent job of recognizing human speech as other human beings can, and do it economically, speech will eventually replace cards, paper tape and even keyboards for communication with computers. But we are not likely to perform speech recognition economically for some time to come. Thus, it is necessary for us to look for tasks where the economics are only a secondary to the problem of getting the task done. It seems to be rather difficult to come up with a task domain in which speech recognition systems can play a useful role and where the cost incurred is justifiable.

We suggest a few task domains that come to mind.

a. Applications that need human control of large numbers of devices where their hands and feet alone are not sufficient, e.g., aircraft and spacecraft control.

b. Applications where one can only afford an inexpensive input device like a telephone for communication with the computer, e.g., computer conducted polls and referendums.

c. Applications where the sophisticated control provided by the computer is necessary, but the human being in the

---

* The research reported here in was supported in part by the Advanced Research Projects Agency of The Office of the Secretary of Defense under contract F 44620-67-C-0058 and monitored by the Air Force Office of Scientific Research.

[1] See Lea (1968) for an optimistic viewpoint on the value of voice communications with computers.

loop is not able to key-in the necessary data fast enough to keep up with the rapidly changing situation, e.g., air traffic control problems.

d. Scientific problems such as automatic protocol analyses which are used to model human problem solving behavior. Here we have a limited task domain in which free-flowing English is used by the human being to describe his problem solving behavior permitting us to construct the necessary semantic model, namely the problem behavior graph, which can then be used to predict what the speaker is likely to say next.

Speech terminals are not likely to replace other input-output terminals in the foreseeable future but are likely to be invaluable in a few specialized application areas just as the graphics terminal had become in computer-aided design problems.

The parallel with graphics terminal is worth pursuing. Although CRTs had been available even before the emergence of the computer, it was not until their use in computer-aided design they had captured the imagination of the computer industry. Since then the use of video-graphic terminals continues to increase each year in many new and unanticipated directions. Once an easily correctable, if not highly accurate, speech terminal becomes available it appears possible that it will be used in many presently unanticipatable ways. This seems inevitable, if for no other reason than that speech is the universal and natural mode of communication. On the other hand, the ultimate acceptance of speech input terminals in day-to-day use is likely to be limited by the cost.

The cost per bit is likely to be much higher for speech input terminals than for discrete devices like a keyboard for the following obvious reasons. Firstly, some processing will be required to discretize the continuous speech signal. Secondly, since the resulting input string (of pseudophones) can never be guaranteed to be error-free, the string interpreters in the monitors and compilers will have to have better facilities for error detection and correction. The cost of performing these functions will require non-trivial amounts of processor time in contrast to present day I-O devices. However, if the present trend continues and the processors become less and less expensive while the cost of mechanical I-O devices remain steady, the cost of a speech terminal may not be as exorbitant as it might seem at a first glance.

The complexity of present speech processing algorithms indicates that a speech terminal is likely to be another peripheral digital processor rather than a hard-wired device. The cost of this peripheral processor will depend on the performance expected of it. To be more specific the cost per bit, while using a speech terminal, may be reduced by relaxing any of the following performance measures, e.g., accuracy, response time, size and structure of the vocabulary, and size and structure of the language as illustrated by the following remarks.

1. Cost can be reduced by lowering the expected accuracy. Computational effort appears to grow exponentially with the required accuracy. Our experience indicates that almost any approach to speech recognition can expect to achieve 80% accuracy. Twice as much computational and research effort seems to be required to increase the accuracy from 80% to 90%, twice as much again to go from 90% to 95%, and so on.

2. In applications where response is not critical the cost can be reduced by using a less expensive processor.

3. Larger vocabularies will require more memory to store the lexicon of acoustic descriptions and correspondingly more time to search the lexicon.

4. Discrimination among phonetically similar words (spit, split, slit) requires substantially more computational effort than between phonetically dissimilar words. Thus, the cost can be reduced by carefully choosing the vocabulary. This might occasionally require going to the extreme of coining new words not in the language.

5. Phonemic ambiguity among words can often be resolved at a higher level if two similar words do not occur in the same syntactic or contextual position. Thus by suitably modifying the structure or the complexity of language one can reduce the cost.

These considerations indicate how systems can be tailored to suit the needs of any specific application and are also useful for evaluating the effectiveness of many different approaches to speech recognition.

The overriding factor governing the cost, usability, and availability of a speech terminal will be the progress we make in research over the next decade. If we attempt to extrapolate from our experience of the last two decades we find that the future is very bleak indeed. When one looks for the reasons for the slow rate of progress of the last two decades in speech recognition research it becomes obvious that investigators have in general grossly underestimated the complexity of the problem. In the face of unexpected difficulties, many left the field after having traced the same uncertain ground without building on each other's results. Others chose to work on some peripheral undemanding problems where the criterion for success or failure is not as well defined. Some knowledgeable scientists, who might well have made the difference, chose to ignore the problem reasoning that recognition of spoken English with equal facility as a native speaker in a noisy environment seems far away.

Lacking the long term intensive problem oriented research that any complex problem needs, progress has naturally been slow.

The slow progress in speech recognition has also been due to inadequate models. Attempts to force speech recognition into the simplistic mold of a "feature extraction-classification" paradigm of classical pattern recognition have met with partial success in the recognition of digits and other very small vocabularies (Talbert et al, 1963; King and Tunis, 1966). But with large vocabularies and connected speech this paradigm is either unusable or tends to become a brute-force technique. At the other extreme, models such as 'Analysis-by-Synthesis' (Stevens and Halle, 1964), have not progressed much beyond the proposal stage for a number of reasons, not the least of which is that synthesis has proved to be no easier than analysis.

Inadequate technology has also been responsible, in part, for the slow progress. Before the availability of appropriate computer systems, attempts by Fry and Denes (1959) and Sakai and Doshita (1963) to build speech recognition machines were abandoned after limited success. The main reason appears to be that hardware modification and checkout of a new idea often requires many man months of effort and at the end one may have to un-modify the system since the attempt did not succeed. Even now, most speech research groups are limited to the use of small dedicated computers which make it difficult to experiment with complex models. When larger computer systems were used, (Bobrow and Klatt, 1968), the inability of the monitors to handle large data rate real-time requests has forced researchers to use limited, prerecorded data sets thereby making it difficult to measure the performance of the system in a realistic situation.

The HEAR (Highly Efficient Audio Recognition) system being developed by the author and his colleagues at Carnegie-Mellon University does not suffer from some of the above disadvantages. This system uses a large time-shared PDP-10 computer with real-time facilities and is based on an "analysis-by-learning" model. The acoustic features required for comparison are abstracted from actual utterances and stored in a lexicon, thereby eliminating the need for the specification of an apriori model of speech production required by the "analysis-by-synthesis" approach. A presently working program, based on this model, was written by Vicens (1969) as part of his doctoral dissertation and is capable of close-to-realtime interaction.

THE HEAR SYSTEM

The main aim of the HEAR system is the recognition of connected speech of languages of about the same complexity as the present day computer languages, with an efficiency approaching the human perception of speech. This goal was chosen because it separates the problem of connected speech recognition from the problem of dealing with the idiosyncrasies of the English language and because this appears to be the most difficult subproblem in speech recognition which can be undertaken with some hope of achieving the goal in less than ten years. The requirement of connected speech was imposed because the system would be of very limited use if the speaker had to pause between words. It is assumed that languages like Fortran and Algol would be awkward for speaking to computers and that each user would specify his own language to suit the needs of his problem. The system is expected to handle vocabularies of around a thousand words without too much difficulty.

A critical requirement of the HEAR system is that it should equal human performance in at least a limited language situation. The time for recognition should be no more than the time for saying the utterance. Furthermore, most of the analysis is expected to take place concurrently while the command is being uttered so that the task requested may be performed immediately following the utterance. While a large number of approaches to speech recognition have been suggested, most of them seem to ignore the question of efficient recognition. The literature abounds with brute-force methods of questionable value.*

The requirement of highly efficient recognition of connected speech of non-trivial vocabulary makes our approach significantly different in many ways from various short term attempts at speech recognition, as will be demonstrated in the rest of this section. Classical methods of pattern recognition, such as the use of a metric in a multidimensional space partitioned by hyperplanes, are not easily extendable for analysis of complex sequence of sounds which may be part of a spoken message. The structure of the message and the interrelationships among the sounds of the message are important factors. Even in those parts of the analysis where classification is required, such as the comparison of part of the utterance with the entries in a lexicon, what seems to be more important than classification is the selection of a few relevant candidates from the lexicon by heuristic and associative addressing techniques. Similar comments apply to analysis by synthesis approach. Clearly one does not want to synthesize acoustic representations of many different utterances each time in the hope that one of them will match the incoming signal.

Analysis by Learning

Since we have no satisfactory model of human speech perception, we have found it necessary to let the machine formulate its own model from a training set of words and sentences of the language to be recognized. This implies that we must provide the system with the necessary data structures and processes which are able to abstract the relevant information from the training set. In our present system many of the thresholds and heuristics which can conceivably be abstracted from the training set are 'built-in' as is the syntactic and contextual information about the language to be recognized. We expect that, at some later stage, the system will be able to modify its thresholds, heuristics, syntax, and contextual information based on past experience.

---

*The reader can verify the validity of this statement by applying the performance measures discussed earlier to the long list of references given in Pierce (1969).

Learning in the present system is restricted to the construction of a lexicon of acoustic descriptions associated with concepts (or print names). Learning, in our limited context, is defined to be the process of modification of the data structure of the lexicon by a previously unknown acoustic structure. Recognition, then, is the comparison of the incoming acoustic description with the various entries in the lexicon. By organizing the lexicon in terms of the gross sequential structure of the utterance, comparison can be limited to only those entries in the lexicon that have similar structure. The similarity between the parameters of corresponding segments of the incoming utterance and an entry in the lexicon is measured in terms of the similarity score with the range of 0 to 100 (very dissimilar to identical). Recognition is defined to be the discovery of that entry in the lexicon which has the highest score exceeding a given threshold.

The analysis performed is the same whether the system is learning or recognizing. In both cases the system searches the lexicon to see if there exists an acoustic description in the lexicon that corresponds to the presently analyzed part of the incoming utterance. If the search fails or if the result is wrong, a new entry is made into the lexicon. Thus learning is treated as a special case of recognition. Of course it is also possible to direct the system to 'always learn' or 'only recognize'.

The HEAR system does not attempt to model human learning of speech. The basic structure of the lexicon is much more analogous to rote-learning. However by making it possible to associate several names to various parts of the same acoustic description and vice versa a much richer, though complex, memory structure is obtained than by a one-to-one mapping of names and acoustic descriptions.

Note that the learning mechanism proposed here is very different to the learning implied by the use of perceptron-type of devices. We do not propose to connect a parameter extractor to a learning net and expect it to adapt itself. The emphasis here is on the development of a sophisticated data structure capable of acting as an associative net. Thus, once certain gross characteristics of the incoming utterance are known, they can be used to localize the search to some small subpart of the associative net. Also note that the input to the system is the labeled pseudo-phonemic segments and not the raw acoustic signal or the output of a bank of filters.

The HEAR system has at least five different phases: parameter extraction, segmentation, sound classification, sentence analysis and word boundary determination, and word recognition. These operations on the incoming utterance are not normally intended to be performed in sequence. We expect that they will act as a set of coroutines with feed back from higher levels guiding the search at lower levels. In each stage the system has to use phonological, syntactic and contextual constraints to reduce the search. The rest of this section discusses the problems and functional characteristics of various phases of the system. Those who wish to find the implementation details of the systems are referred to Reddy (1967), Reddy and Vicens (1968), and Vicens (1969).

Parameter Extraction

One as-yet-unresolved problem that has attracted more than its fair share of attention is the search for the so-called 'acoustically invariant' parameters of speech (Lindgren, 1965). Although certain dominant features like formants were discovered, it was found that most of these dominant features could not be counted on to be present for every speaker, or even the same speaker in a slightly different context. It appears that, if we consider phonemes to be made up of bundles of features, the presence of a majority of these features is sufficient for the perception of the phoneme. So much so it sometimes happens that two completely disjoint bundles of features are perceived as the same phoneme by a human auditer. Researchers who hope to discover the features relevant for analysis by synthesizing speech should beware. Just because they have succeeded in synthesizing, say, a single phoneme /k/ they should not expect to find the same set of features in every phoneme /k/. These considerations have led us to abandon the search for acoustically invariant features of speech and build a system that is not critically dependent on the presence or absence of any single feature.

The other main as-yet-unresolved problem is 'what type of analysis should we use to extract the parameters?'. After many years of experiments with zero-crossing analysis, spectral analysis, formant analysis, polynominal fitting, etc., my somewhat surprising conjecture is that 'it does not matter'. The main problem is that in day-to-day speech the acoustic signal provides only part of the information. The rest is provided by the listener's own contextual and extra-linguistic abilities. No amount of sophisticated analysis can discover features that are not present in the original signal. So much so it seems irrelevant to fret about determining a frequency very accurately when that component might very well be absent the next time. It is my conjecture that, for most recognition tasks, it does not matter what type of analysis is used as long as the results of the analysis are consistent. The syntactic and contextual information is usually sufficient to resolve ambiguities arising from the variability in parameters. When the language to be recognized becomes more complex, such as two phonemically ambiguous words occurring in the same syntactic position a careful look at the acoustic signal might be needed. Useful but less dependable features are extracted in our system only when they are absolutely required.

At present we use as parameters zero crossings and amplitudes in six frequency bands in the audible frequency range sampled every 10 milliseconds. We have found this to be a reasonable compromise between high data rate 40 channel filter bank and low data rate amplitude and zero crossing measurements of the original signal.

Segmentation

Figure 1 illustrates the machine segmentation of the utterance 'how now brown cow'. Note that the diphtong /au/ takes on different shapes in different contexts, illustrating one of the reasons why consistent segmentation is difficult to achieve.

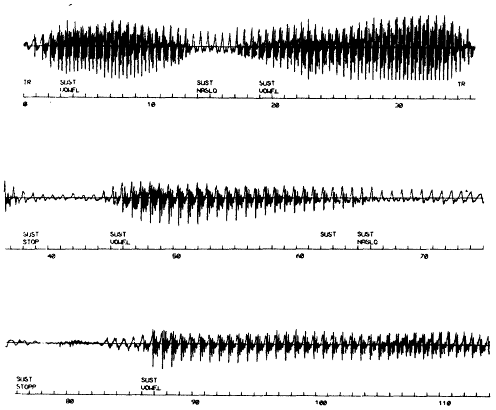


Figure 1.
The many faces of 'ow' in 'How now brown cow'.

Another instance of 'how now brown cow' might result in a different number of segments even when it is uttered by the same speaker in the same environment. These difficulties in obtaining consistent segmentation have led many investigators to look for approaches to recognition which do not require segmentation of the acoustic continuum into discrete parts. The analysis-by-synthesis approach is one such. Segmentation-free recognition has so far proved to be usable only in very small limited vocabulary situations. It is usually time-consuming because it does not lend itself to techniques for reduction of the search space.

That we need segmentation for the analysis of connected speech is obvious. The question is what type of segmentation should we use: phonemic, syllabic or morphemic segmentation? Present linguistic definitions are usually subjective and almost impossible to specify in an algorithmic form suitable for use by a computer. Many investigators just locate well defined boundaries such as unvoiced fricatives and stops (Hughes and Hemdal, 1965; Denes and von Keller, 1968). The main disadvantage with this approach is that sentences such as 'How are you' would then have to be treated as a single segment thereby complicating subsequent analyses. In our system we find we need all the three concepts of phoneme, syllable, and morpheme. Hence we have defined the concepts of pseudo-phoneme (a collection of adjacent 10 ms segments with similar acoustic characteristics) and pseudo-syllable (a collection of phonemic segments containing one and only one local maximum of amplitude) to be suitable for machine segmentation. A hierarchical segmentation procedure for obtaining pseudo-phonemic segmentation is given by Reddy and Vicens (1968).

Segment Classification

Classification of segments into phoneme-like categories, while unreliable because of the variability of parameters discussed earlier, is often useful for the generation of an ordered candidate list by associative addressing techniques. The difficulties in obtaining reliable phonemic classification has led us to generate segment descriptions in terms 10 to 15 supra-phonemic categories. Such classification, while not in itself complete, is useful in describing the gross characteristics of a segment. This description can be used in minimizing the search space in word boundary analysis and word recognition.

Word Boundary Analysis

Determination of word boundaries in connected speech is by far the least understood of all the problems. The apparent preoccupation of most investigators with the acoustically invariant parameters of speech has been responsible for the lack of progress in the subsequent problem areas. Our own limited investigations show that this is likely to be the main bottleneck in the analysis of connected speech. Two main sources of the difficulties are the substantial modification of acoustic characteristics of a word in various contexts and the word boundary ambiguity problem, e.g.,'ice cream', vs. 'I scream'. We are presently using a temporary expedient which requires careful selection of the syntax and the vocabulary of the language so as to minimize these difficulties.

Word Recognition

A main problem in word recognition is the correction of segmentation errors. When two utterances of the same phrase by the same speaker results in a different number segments, the question arises as to which segment corresponds with which, so that proper comparison can take place. This is known as the segment synchronization problem in speech recognition (Reddy, 1969). It is similar to sequential decoding in the presence of errors in communication theory. In our system a mapping procedure determines the correspondences between segmental descriptions. Dominant segments corresponding to vowels and fricatives are mapped first. The remaining segments are then mapped on the basis of similarity of parameters.

Another problem in word recognition is the formulation and specification of various heuristics to reduce the time per recognition. In various word recognition experiments we found that as the vocabulary increases from 50 to 500 words, the time spent in searching the lexicon increases from 50% to 90% of the total time for recognition even though the search procedure uses several heuristics for the reduction of search space. This fact reiterates my earlier comment that there exist many other important problems in speech recognition research besides feature extraction and classification. Word recognition, then, requires the development of efficient procedures for the search of the lexicon, and these become critically important as the vocabulary increases. The following heuristics, used in our system, illustrate the type of devices that are helpful in reducing the search.

1. The data representation in the lexicon is arranged so that only those entries of the lexicon which contain the same number of syllables and unvoiced fricatives as the incoming utterance are considered first.
2. The search is terminated when a candidate obtains a high enough score, say 95% similarity.
3. If a candidate has a different global structure, i.e., if the sequential similarity constraint is not satisfied, the candidate is rejected without any further processing.
4. The candidates are ordered so that candidates with similar vowel structure are considered first.
5. If the stressed vowel is significantly different, then the candidate is rejected.
6. If a candidate obtains a high score but not enough to terminate further search, the candidate list is reordered so that all other phonemically similar candidates are considered first.
7. If no candidate in the initial list obtains a high enough score, say > 80%, then an attempt is made to transform the incoming description to correct for possible errors.

The word recognition system developed by Vicens (1969) obtains 92% to 98% accuracy for a single speaker depending on the noise and amount of learning. For multiple speakers the accuracy is around 80% to 85%. For a given accuracy the recognition time is a function of the size of the vocabulary and varies from 6 times real time for a 50 word vocabulary to 30 times real time for a 500 word vocabulary on a PDP-10 system. We seem to be a factor of $\approx$ 50 away from our goal of real time recognition for a 1000 word vocabulary.

UNSOLVED PROBLEMS

Of the many unsolved scientific and engineering problems in speech research I shall restrict myself here to those problems that are likely to be critical once the speech input terminals leave the laboratory environment and the tender loving care and protection of their creators. These concern the variables governing the characteristics of the speaker, the terminal, distance between the speaker and receiver, noise, etc.

In speech, characteristics of an utterance vary not only from speaker to speaker depending on the sex, age and physical condition, but even for the same speaker depending on his emotional state at the time of utterance. In our experiments we found that utterances of a speaker irritated by the mistakes of the machines are anything but normal. Speaker variability is at present minimized in our system by training for individual speakers and by requiring the speaker to be cooperative. The main limitation of this constraint is that every speaker must train the system before he can use it reliably. Mr. Erman of our group is attempting to formulate techniques for speaker normalization. Determination of differences and similarities of the characteristics of various speakers is one of the unsolved problems that is likely to require many man years of sustained effort.

The human ear can detect sounds between 50 to 20,000 Hz of sound intensities within the 0 to 110 decibel range with the smallest detectable sound resolution of 10 to 20 ms (/l/ as in slit as opposed to sit). Most voice input systems to computers have a 100 to 5000 Hz frequency response with a 40 decibel range ($\approx$ 8 bits of resolution) with a sound resolution of 10 ms. The lower frequency response results in an occasional confusion between fricatives /f/, /θ/, and /s/. While it is within the capabilities of the engineering technology to build a receiver of any desirable characteristics, there has not been much effort to determine the optimal characteristics of a receiver for speech terminals satisfying the conflicting requirements of low bit rate and wide dynamic range.

The distance between the source and the receiver is also likely to be a problem for speech terminals. A microphone held too close to the lips also records the expiration after the utterance giving the illusion of an extra sound, and when held too far results in the loss of resolution with an additional problem of lower signal to noise ratio. Reliable discrimination between speech and expiration is proving to be difficult (note /h/ sound in English is a special case of expiration). Development of the mini-microphones which can be attached to collar might minimize this variability.

By far the most difficult problem for speech input terminals is likely to be the discrimination between the speech source and various noise sources. While some of the attempts at noise reduction (such as the design of directional microphones) are acceptable, others (such as putting the speaker in a noise-free acoustic chamber) would not be acceptable for use with speech terminals. Some of the sources of external noise for speech terminals are: air conditioning and fan noise, printer noise, other speakers in the room, 'Hmms' and 'Hahs' and clearing of the throat of the speaker himself, etc.

There is no simple unique solution to all these problems. Software solutions to these problems are likely to be difficult and time consuming, and are not compatible with the less than real time recognition requirement for speech input terminals. Social solutions, such as that no one may sneeze or cough in the speech terminal room are not likely to work either. Thus it is imperative that speech terminal designers design their system so that an occasional error cannot be catastrophic. Further, it should be possible to correct the system with minimum of effort. One possible solution is to couple the speech input terminal with a CRT for error detection and correction. In a real-time environment the commands would appear on the CRT as they are uttered by the speaker permitting him to immediately verify and correct the command in case of an error.

Unlike other I/O devices, the initiation and termination of I/O for speech terminals is data dependent. There are some devices in the market for the detection of the presence or absence of speech. However, since these are amplitude activated, they are noise sensitive and cannot yet be activated by low amplitude sounds, such as stops and fricatives. Development of a more sophisticated device will be necessary to minimize unnecessary interrupt processing by the computer.

CONCLUDING REMARKS

In this paper I have tried to outline a number of factors affecting the cost, utility, structure, and engineering of speech input terminals for computers. In particular, it is not enough to just measure the accuracy of a proposed algorithm, but one must consider all the relevant factors, e.g., accuracy, response time, vocabulary size, complexity of words, and complexity of language. All of these will affect the cost, utility and structure of a speech terminal.

Seymour Pappert of M.I.T. once said, while commenting on the disappointing rate of progress in robotics research, that if we were to think that building a robot requires any less effort than putting a man on Mars, we would be sadly mistaken. Since we can produce people much less expensively it is very unlikely, at this time of shifting national priorities, that the billions of dollars in funding required for the research, development, and engineering of a robot will be forthcoming. Speech perception, being a difficult part of robotics research, is likely to fare no better. In view of the limited resources available for this type of research, it is essential that we avoid duplication of research, choose research goals that are likely to be of lasting value, avoid working on inconsequential peripheral problems, and develop a close cooperation between various interested research groups.

ACKNOWLEDGEMENTS

This speech research project was initiated by the author at Stanford University in 1964, following the lead and inspiration provided by Dr. John McCarthy. Since then, several graduate students have made significant contributions to the project, both at Stanford and Carnegie-Mellon Universities. Particular mention should be made of Pierre J. Vicens who wrote our first working real time word recognition program and Dr. Morton Astrahan, Lee Erman, Gary Goodman and Richard Neely who are developing interesting techniques for machine learning, speaker normalization, language design and conversational systems, respectively. The author is indebted to Dr. Allen Newell for many thoughtful comments on a preliminary version of this paper.

REFERENCES

1. Bobrow, D. G. and Klatt, D. H. (1968), "A Limited Speech Recognition System", Proc. FJCC '68, 305-318.

2. Denes, P. B. and von Keller, T. G. (1968), "Articulatory Segmentation for Automatic Recognition of Speech", Proc. International Congress on Acoustics, Tokyo, 11, B143-B146.

3. Fry, D. B. and Denes, P. B. (1959), "The Design and Operation of a Mechanical Speech Recognizer", J. British IRE, 19, 211-229.

4. Hughes, G. W. and Hemdal, J. F. (1965), "Speech Analysis", Tech. Rept AFCRL-65-681, (P137552), Purdue University.

5. King, J. H. and Tunis, C. J. (1966), "Some Experiments in Spoken Word Recognition", IBM J. of R. and D., 10, 1, 65-79.

6. Lea, W. A. (1968), "Establishing the Value of Voice Communication with Computers", IEEE Transactions on Audio and Electroacoustics, AU-16, 2, 184-197.

7. Lindgren, N. (1965), "Automatic Speech Recognition", IEEE Spectrum, 2, 3, 114-136.

8. Pierce, J. R. (1969), "Whither Speech Recognition", J. Acoust. Soc. Am., 46, 4, 1049-1051.

9. Reddy, D. R. (1967), "Computer Recognition of Connected Speech", J. Acoust. Soc. Am., 42, 2, 329-347.

10. Reddy, D. R. and Vicens, P. J.(1968), "A Procedure for Segmentation of Connected Speech", J. Audio Engr. Soc., 16, 4, 404-412.

11. Reddy, D. R. (1969), "Segment Synchronization Problem in Speech Recognition", J. Acoust. Soc. Am., 46, 1, 1, 89 (abstract).

12. Sakai, T. and Doshita, S. (1963), "The Automatic Speech Recognition System for Conversational Sound", IEEE Trans., EC-12, 835-846.

13. Stevens, K. N. and Halle, M.(1964), "Remarks on Analysis by Synthesis and Distinctive Features", Proc. of Symposium on Models for the Perception of Speech and Visual Form, AFCRL, 1964, Ed. by W. Wathen-Dunn, MIT

Press, 88-102.

14. Talbert, L. R. et al (1963), "A Real-Time Adaptive Speech Recognition System", Tech. Rept. No. 6760-1, (ASD-TDR-63-660), (P133441), Stanford Electronics Lab.

15. Vicens, P. J. (1969), "Aspects of Speech Recognition by a Computer", Ph.D. Thesis, AI Memo No. 85, Computer Science Department, Stanford University.

MANIPULATORS AS TERMINAL DEVICES

Alan C. Kay

Stanford University

ABSTRACT

A "classical" manipulator is a tool for augmenting the use of the hand.

To use it as a terminal device, a computer is inserted between its afferent and efferent boundaries forming a full-duplex linkage. A variety of possible terminals are discussed including stylii, "waldos", micromanipulators, prosthetics, and "Hardimen". Means of providing and normalizing kinesthetic, tactile and other feedbacks are also covered.

The "total terminal" can be created by completely surrounding the user by sensors. The advantages and limitations of this kind of simulation provide a measure of the state-of-the-art of current terminal designs.

INTRODUCTION

A generalized computer aided manipulator is shown in figure 1. The human operator has hand controls, earphones, and sees through a television monitor on the "A" side of the linkage.

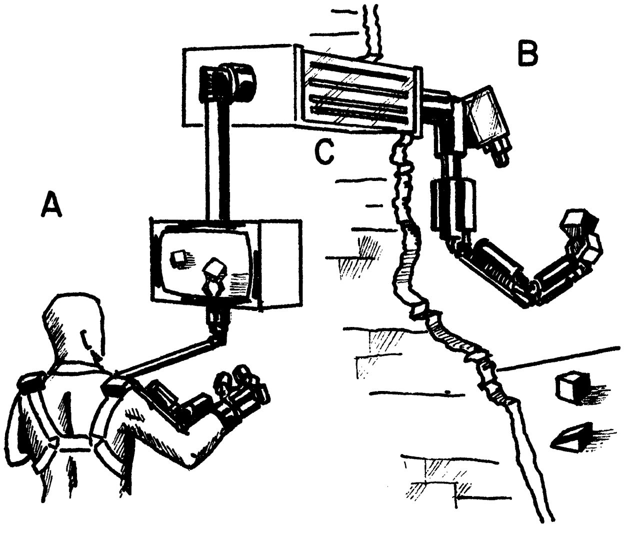


Fig. 1. A Full Duplex Linkage

If this side is omitted the computer must simulate the operator's actions in order to move the manipulator. This is the "hand-eye-ear" problem in Artificial Intelligence research.[1]

A raster of the scene is input as a stream of grey-scale codes to a program which attempts to find shaded regions that represent bodies. Once found, a task such as building a simple house may commence. At no time is human intervention necessary.

If the program is able to recognize objects in the scene (in this case cubes and wedges), it can then build a model for future reference so that the laborious task of analysing the TV raster need not be repeated.

If a sonic burgler alarm or similar device is part of the environmental sensors, the system can be warned of a change with little or no computational cost and a TV scan can again be made to determine what happened.

If the other side ("B") is removed, the computer is used to simulate the environment, feeding back images, sounds, and tactile sensations through the manipulator terminal.

Suppose the environment, is again a collection of cubes. A "halftone" algorithm [2,3,4] can be used to quickly generate a shaded rendering of the objects as a raster of points which is transmitted to the operator's CRT. The same is done for a visual simulation of the gripping device. The nature of most "hidden surface" algorithms will also allow them to detect intersections of the claw with the object . If the rate of intersection is opposed by applying force to the joystick-glove of the human operator, the "object" will be "felt" through the glove.

If "C" is omitted or rudimentary, a common industrial manipulator, such as those used in dealing with radioactive material results.

This paper deals with permutations of scale, organization, and intent of this basic model in a speculative way. Many of these concepts were developed in conversations with Dr. Ivan Sutherland of the University of Utah.

COMPUTER AIDED MANIPULATION

The idea of the modern industrial manipulator was first popularized in a science fiction story by Robert Heinlein in 1938.[5] A "waldo" (named after its fictional inventer) resembled a pair of gloves which could sense and transmit the user's hand movements to a variety of machines ranging from huge industrial cranes to the most delicate of micromanipulators. The waldo gloves were transceivers as well; forces exerted by loading,the temperature of the slave environment, etc. were transformed back into acceptable ranges for the human operator and applied through the gloves. Occasionally television would be used as a substitute for direct vision.

Although no devices as well designed and engineered as these yet exist, the first manipulators for remote handling of radioactive materials were called "waldos" in honor of the basic idea.

The first of these was developed at the Argonne National Laboratory in 1947 [6,7,8]. Both mechanical and electrical linkages were developed. Commercial versions and improvements of the basic devices have appeared over the years [9,10,11]. As with Heinlein's waldos, the Argonne manipulators are "master-slave" devices; the method of control is mimicry through negative feedback servos.

Other forms of control have also been used. The commercial "back hoe", a scoop connected to a tractor, is controlled by a number of one-axis joysticks mounted in parallel and requires considerabe dexterity and coordination to operate. The manipulators used for deep-sea diving [12,13] also tend to be joystick or toggle switch controlled. It is not entirely clear why this is the case since "master-slave" makes use of long developed habits of dexterity without requiring new patterns to be learned which are at least of the order of difficulty of basic piano rudiments.

The first union of a computer and a servo-manipulator occurred in the early sixties when an AMF "waldo" was attached to and controlled by the TX-0 computer at the MIT Lincoln Laboratory.[14] It used only a light sensitive diode for "visual" feedback and depended mainly on tactile sensations gathered during a "grope" for objects. After it had located blocks it could build a stack under program control.

Many of the ideas developed in this work are still current practice. A rather comprehensive model was developed in the computer for holding known positions of objects as well as those areas which had not yet been searched.

Since then the two centers of activity for optically directed manipulation have been Stanford University and MIT [15,43,44].

The early Stanford work involved programs which could interpret a television raster to determine if cubes of various sizes were present in the scene. Once located, block stacking routines would be called to segregate the large blocks from the small [16], obstacles could be avoided [17], and optical feedback could be employed to align a stack of blocks [18].

The voice recognition programs of Reddy and Vicens [19,20], were also coupled to the system to provide vocal inputs that could previously only be given by teletype.

The manipulator used in these experiments (aptly named "Butterfinger") was a powered orthotic brace developed at the Rancho de Los Amigos Hospital. Its failings are legend among the cognoscenti.

Both the MIT and Stanford groups have taken a fling at designing exotic non-anthropomorphic manipulators with little success and much spilled oil. Minsky's group developed a "snake-arm" with nine intertwined joints, while McCarthy oversaw developement of a tremendously powerful (2000 psi, ~200 Hz response), quite dangerous arm (it resides in a little room of its own and truly tests a persons faith in his debugging ability). It has three elbow joints. Neither arm has proven tractable to analysis for easy positioning of the terminal joint although Kahn [21] has shown that a near-optimum control is possible.

The current manipulators in use by MIT and Stanford are a little more prosaic than the preceding devices. However, they both show evidence of greater accuracy and reliability. The new Stanford device ("The Golden Arm"[22]) has a turntable as the staging area which allows the object field to be easily viewed from all sides. Pan, tilt, and focus of the T.V. camera are also servoed and under computer control. Among other things, this allows stereo ranging by binocular triangulation.

The grasping mechanisms of all these manipulators are simple vice-grip hands. The MIT laboratories have also tried a prosthetic device, the "Belgrade" Hand [23], on their arms. The "eye" problems have been so severe that little has been done with hand design. (This subject is covered in more detail in the section on grasping devices).

Dr. Fred Brooks and Joseph Capowski are initiating a project in computer aided manipulation at the University of North Carolina. They expect to completely simulate the environment using graphics and force feedbacks.[24]

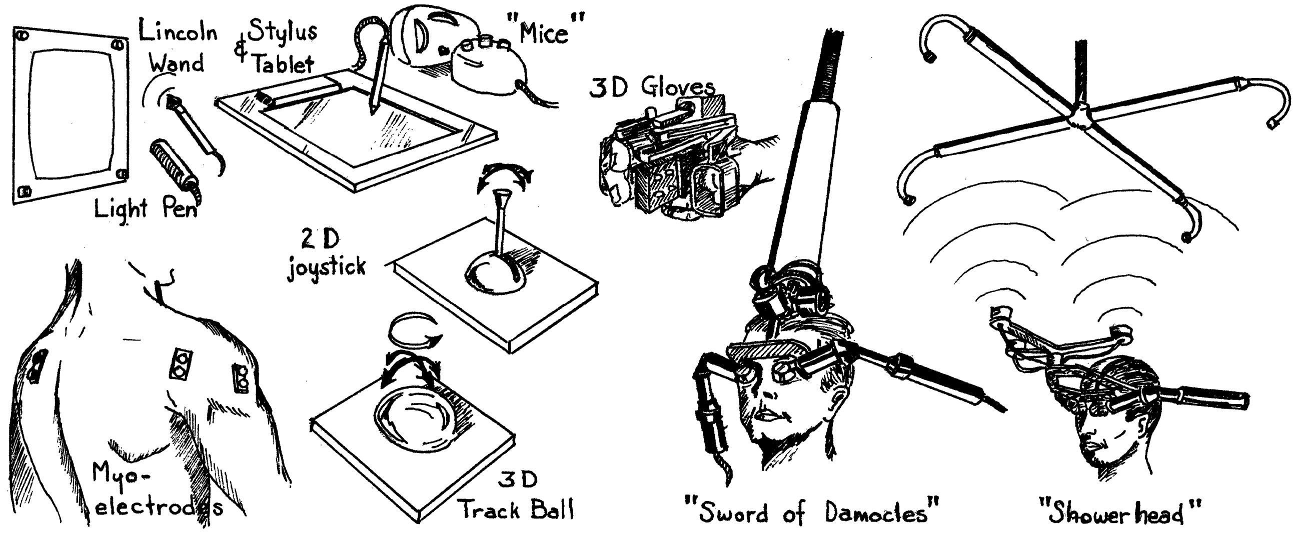


Fig 2, Current Input Devices,

The joystick, gloves, trac ball, "mice" and "sword of Damocles" either use potentiometers with A/D converters or digital shaft encoders for providing angular data, The nature of the mechanical linkage of these devices also allows kinesthetic feedback to be applied through torque motors or selsyns controlled by the computer.

The joystick and track ball have two and three degrees of freedom respectively, The readings obtained from each axis are commonly used as scale factors for position, They may also be spring loaded to return to the zero setting with the readings used as multipliers for velocity or acceleration, The correspondance between axes and translational or rotational motion is usually an implementation decision, If all proper motions of a referential frame in a space are to be controlled, six axes will be required, (see next section),

Input mechanisms that are used in Link trainer-like devices and space navigation simulations are also included in this category,

The light pen, tablet and stylus, mouse, and the joystick, are used to locate a point in a plane,

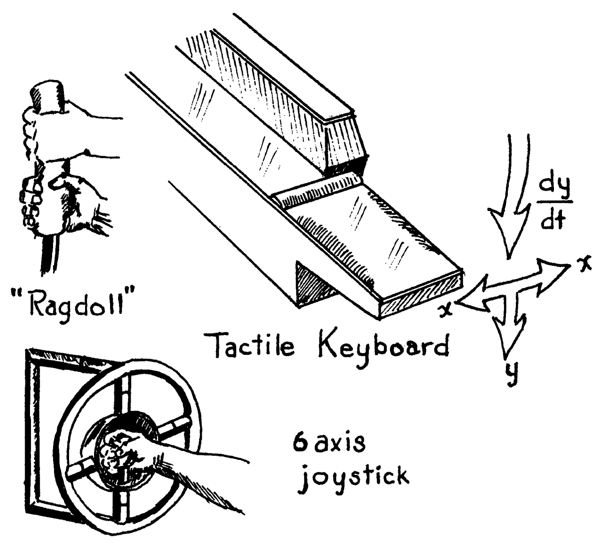
The tablet may measure analog or digital voltages, phase, x-y position of a tracking magnet on the undersurface which is dragged by the stylus, or even x-y ultrasonics through the reference plane, In general, the tablet has been by far the best of the 2-D devices, The light pen gets confused by ambient light, tends to cause the blood to drain out of the user's hand and has very poor resolution, The mouse can be used to point, but is somewhat less than ideal for accurate sketching, [25]

The manipulator "gloves" and the "sword of Damocles" are quite simillar, A typical glove can be used with either hand and consequently yields low resolution signals as to the operators intentions,

The "sword" measures head position accurately in six degrees of freedom and is used in conjunction with half-silvered CRT goggles (one tube for each eye), The military has employed it to superimpose images of flight instruments or a rear (or bottom) view over the pilot's visual frame while he is gazing out the canopy, This allows a great deal of information to be presented without disturbing his concentration,

Ivan Sutherland, then at Harvard, used the "sword" and goggles as a computer display device, A very fast matrix multiplier (a 4 vector times a 4x4 matrix in 10us) provides a real-time transformation from a "room" coordinate system to the "eye" coordinate system,[26,27] When the wearer changes his head position, the "sword" measures the change, the matrix is permuted according to the new data, and the display changes, The user, however, feels that the displayed object is "still",

The Lincoln Wand and the "shower head" are both ultrasonic ranging devices used for gathering three-dimensional data, The Wand was used to "sculpt" and direct three-dimensional displays[28], It proved to be very sensitive to ambient noise aod otherwise hard to control, The "head" uses phase measuring and wavelength counting as well as redundancy to eliminate noise problems, It is still subject to air currents and wall reflections, Sutherland [27] has used it as an alternate to the "sword" with the advantage of not being physically attached to the user, (The "sword" makes the wearer's head feel as though it has about four times normal mass),



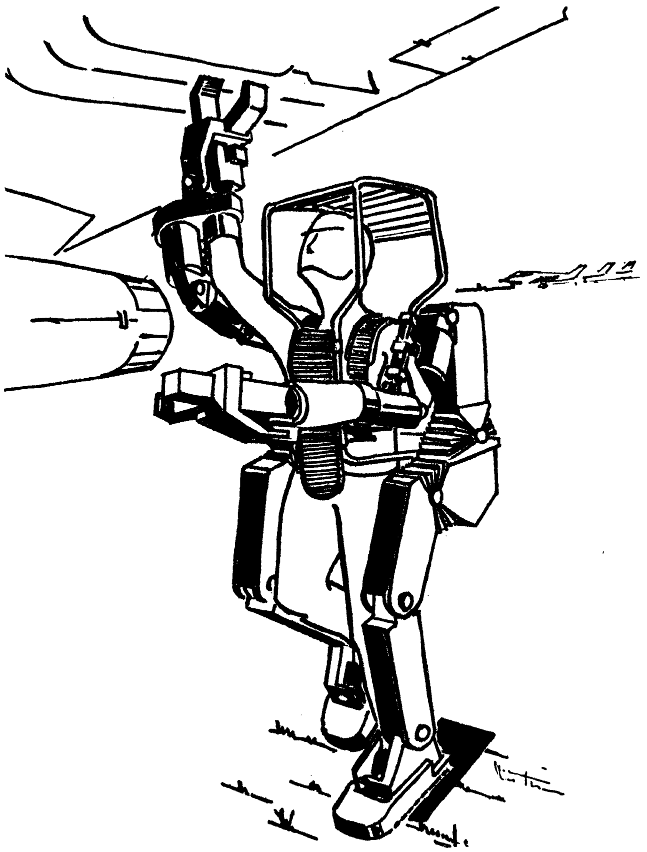
Fig, 3, Future Tactile and Kinesthetic Input Devices

The "suit of armour" has already arrived in a number of forms, The General Electric "Handyman", "Hardiman" [29,30,31], and related projects use an articulated exoskeleton for sensing main joint movements, This controller resides in a powerful, hydraulically controlled "vehicle" which may or may not be anthropomorphic, The linkages for the most part are simple servo loops and thus do not really enter into this discussion,

The user actually dons an inner exoskeleton which is the master pantograph, The outer, more massive frame attempts to "keep away" from the inner one and thus will mimic the wearer's movements, In the sense of this paper, the computer should be inserted between the inner and outer shells, The outer frame may be directed by the computer resulting in an anthropomorphic "robot", or it may be discarded allowing the inner skeleton to be used as an I/O device for graphic and kinesthetic simulation,

Tactile sensations to the fingers and arms can be provided by pneumatically inflated bladders in the gloves, Temperature feedback is supplied by resistance wires in the "skin" of the device,

Suppose now that the Sutherland goggles are fastened to the master pantograph so that they share the same reference frame as the gloves, This results in an interesting device, The "slave" may exist or be a simulation; either case can generate fruitful results,



Fig, 4, "Hardiman" (c) General Electric Corp,

The prehensile device could be simulated through the display or might be the user's own hands, In either case a hidden line algorithm (the main parts of which can be done by the Sutherland hardware) can be used to detect intersections of the simulated object with the projection of the user's hand in "room" space,[4]

Force feedback can then be applied by torque motors resulting in the user sensing a mass through his gloves,

The field of music can also benefit from manipulation research, The number of simultaneously controlled parameters by an accomplished musician is far larger than any current computer input device, It is not strange that most musicians find computer terminals a bit "dry",

A keyboard in which each key is monitored for vertical position (on-off as with an organ), vertical velocity (as with pianos and tracker organs), vertical force (as with the clavichord "Bebung"), AND right or left "strain" (found on no existing keyboard instrument), The velocity curve can be used to modulate initial transients (strike tone), Vertical force can initially set (and later reset) the volume level, Right or left strain may be used for portamento (slurring the pitch up or down) to produce vibrato effects and expression usually associated with the violin family, A device even less committed to a preconceived use is the "ragdoll",[32] It is simply a form covered with strain gages which can be kneaded by the user, The semantics of each spot can be determined by the interpretation program,

MICROMANIPULATORS

These devices come in many forms, A common example is the pneumatically driven micromanipulator for coarse to medium dissections, It uses orthogonally mounted bellows as force transducers which are compressed by a joystick,

The movements of the joystick are reduced in a 90 to 1 ratio and transmitted to the dissection stage, Other devices use moter driven micrometer stages to allow fine dissections with a minimum of play and backlash,

Ivan Sutherland has suggested a method for adding kinesthetic feedback to these devices, A multiocular 'scope is used, allowing a TV camera to also look at the field,

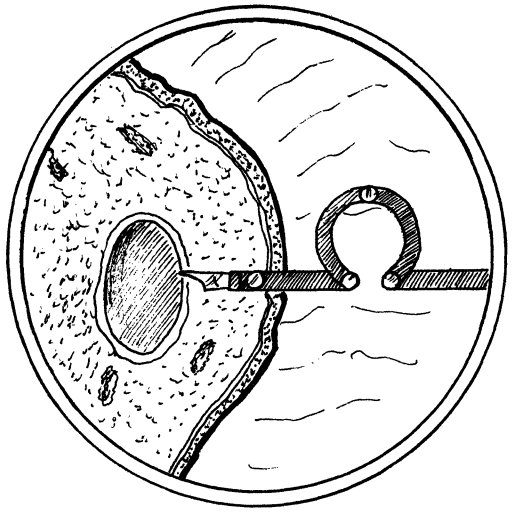


Fig. 5, A Force Sensing Microtool

The dissecting tool is bent into a horseshoe shape as shown and has 4 highly reflecting mirrors which can easily be recognized by a program looking at the video raster, Force exerted against the tool by the specimen will distort the shape of the tool, The program will recognize the distortion and will attempt to return the spots to their rest configuration by opposing the user's motions at the input side through bellows or selsyns, The sensation to the operator is that of "feeling" the specimen through the joystick,

PROSTHETICS

Approximately 44,000 people are fitted with artificial limbs in the United States each year, and it is estimated that there are 300,000 amputees living in the country at this time [33], Unfortunately, even with this pressing need, prosthetic and orthotic (limb bracing) technology has tended to lag for a number of reasons,

One of these is the problem of supplying and transporting sufficient power to allow a prosthesis to do a reasonable amount of work, Recent advances in battery technology due primarily to the space effort have for the first time made electric power feasable for artificial limbs, The technique of pulse width modulation of a constant voltage power supply for DC motors allows their torque and frequency response to approach that of hydraulic systems [34],

Another, more serious, problem is that of communicating the wishes of an amputee to his prosthetic device, For the last decade prostheses have been built which pick up and amplify electric signals generated in feeble muscles or muscles in stumps and thereupon control braces and artificial limbs[35], The advent of the integrated circuit computer and microminiature electronics now reveal the possibilities for vastly more sophisticated control elements between the electrical pickups and the actuators in the prostheses,

A number of powered prosthetic hands have been developed and a few are in clinical use [23], The best of these suffer from the limited myoelectric control provided, Typically, two sets of electrodes are employed to pick up signals from a muscle and its corresponding antagonist, The signals are then rectified to provide a simple binary opening and closing action, More signals could be picked up, but crosstalk between muscle fibers make discrimination of signal patterns a problem, The computing element in presently available hand protheses is the simplest of analog devices and is by far the weakest link in the system,

A more sophisticated use of myoelectric sensing has been developed at the Moss Rehabilitation Center of Temple University Medical Center by Finley, Wirta and Taylor[36,37,38] as a four year project,

One of the design goals was to build an artificial arm which could take its direction from the wearer's muscular volitions without a great of training, Myoelectrodes were attached to a normal subject in order to find a weighting scheme which could isolate movements in each of the basic axes, Somewhat to their surprise, they found a strong seperation even when muscles close to the spine were sensed,

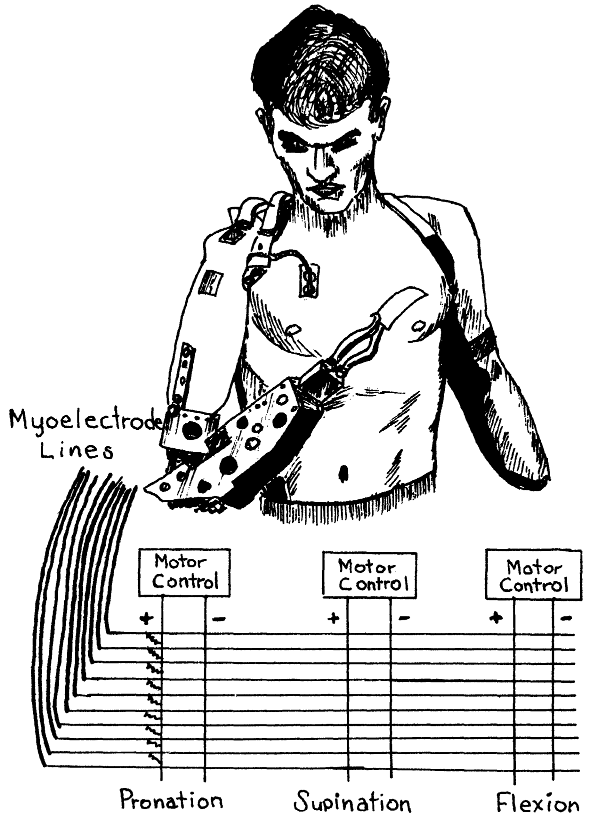


Fig. 6. The Temple Arm

It uses 10 myoelectric sensors distributed on key areas of the rhomboid, trapezius, deltoid and pectoral muscle complexes. Finley, Wirta and Taylor found that even such distal motions as rotations of the wrist have precurser flexions in muscles much closer to the spine. The signals for the three movements identified in the drawing are orthogonal enough to be separately identified by the resistive weighting matrix.

Because gain is essentially fixed, the signal to noise ratio becomes a problem when small precise movements are attempted; the arm will tend to "hunt" somewhat when the user is trying to hold it still. Most of these problems can be alleviated by replacing the analog elements with a small digital computer distributed over a few chips. State information can then be retained and more extensive pattern matching done resulting in very smooth performance of the device...a very important consideration in this case where confidence is most needed.

GRASPING DEVICES

Almost all of the current work being done in grasping device design is in the area of human prosthetcs. Most industrial manipulators, as well as industrial "robots" such as UNIMATE and VERSATRAN [39,40], use either a simple parallel jaw hand or a non-anthropomorphic tool chuck.

The "Belgrade Hand" is a good example of the current state of the art in prosthetic hand design. Its main features are articulated (but ganged) fingers which close around an object, and a palm sensor to tell a limited amount of integrated circuit logic whether the object in question is being picked up with the fingers or encircled by the palm.

Many prosthetic hands lack dexterity, strength, and speed simply because they are designed to fit the standard wrist chuck-joint. Unlike human beings, this prevents the forearm from being used as a site for finger and thumb control mechanisms. So far, this practise has carried over into the sophisticated arm developement area also. The Temple Arm uses its forearm area to control flexion and pronation leaving no room for hand actuators.

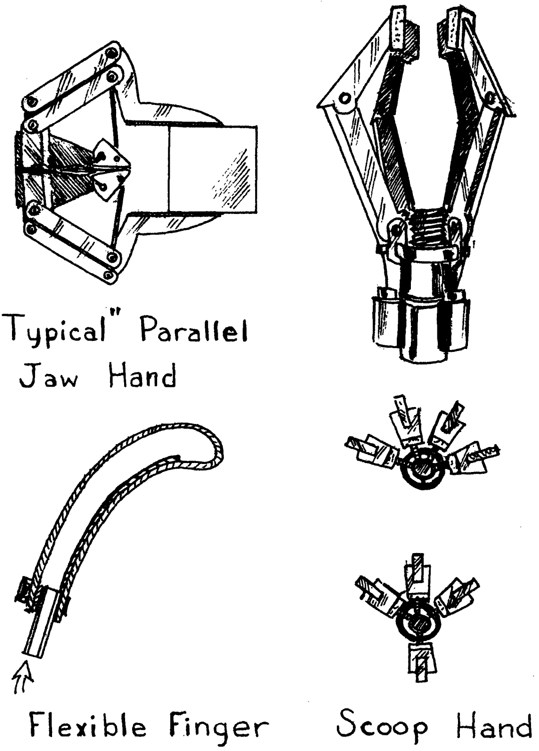


Fig. 7. Current Devices

Examples of current designs under consideration are the "scoop hand" and "flexible finger hand" designs now being evaluated at Stanford.

At the expense of a fair amount of mechanical complexity, the "scoop hand" offers a variety of possible configurations as illustrated in the figure, Both rapid and vernier control of finger closing is provided by the threaded central shaft, An annoying difficulty is presented by the desire to have the radial finger configuration also under servo-control; this can be done but only with considerable sacrifice of simplicity and reliability,

The "flexible finger" is simply a blind hose which is charged with gas under pressure, An extra thickness on one side of the hose will cause the finger to curl inward as the pressure is increased, The actuation is simple enough to allow fingers to be clustered in a variety of ways to suit the current need,

Of course, there is no reason why the output device of a computer aided manipulator should be "hand-like", Some problem areas might be alleviated by using techniques similar to those now used for numerically controlled tools, An example is the Kearney and Trecher "TURN-12" [41], a versatile NC machine shop which uses a computer controlled magazine of tools with servos to change from one tool to another,

The construction industry's bucket, crane and scoop configurations also suggest themselves as possible models,

An alternative to the idea of grasping is offered by the conformable suction-sponge, The vacuum is applied through many small holes in a flexible medium to allow enough of the surface area of the object to be used, Although this allows fairly general holding abilities, a disadvantage is having to use objects that are basically convex and also strong enough not to be destroyed by the suction,

THE "TOTAL" TERMINAL

This "final" device appears in a number of guises, Aldous Huxley's "feelies" in Brave New World describe a total output device in which every sensation is communicated to the audience, Sutherland[42] envisions a "display" in which simulated bullets are dangerous to the operator, The myoelectric sensors already described, coupled with other advances in bio-electrical control and transplantation may indicate a different direction, where the computer terminal becomes an integral part of the user's bio-envelope, CYBORGS are almost inevitable; their appearance has been delayed by the (so far) nonexistence of suitable electronics-to-nerve transducers which will eliminate the need to use physiological sense organs, Transducers are beginning to appear as innocent prosthetics for damaged tissue but may eventually replace healthy tissue as a superior medium for sensation, The man of the future may become a voluntary amputee or have jack plugs to major ganglia in order to run large machine tools or perform delicate micromanipulations mediated by computer,

REFERENCES

1, McCarthy, J,, Earnest, L,D,, Reddy, D,R,, Vincens, P,J,, "A computer with hands eyes, and ears", Proc, FJCC 1968, pp, 329-338

2, Wylie, C,, et al, "Halftone perspective drawings by computer", Proc FJCC 1967,pp,49-58

3, Warnock, J,, "A hidden surface algorithm for computer generation of half-tone images", Tech, rep, 4-15, C,S, Dept,, Un, of Utah, June 1969,

4, Watkins, G,, "A hardware compatible algorithm for generating visible surfaces from 3-dimensional data", Tech note Universty of Utah, Computer Science Dept, March 1970,

5, Heinlein, R,A,, Waldo and Magic Inc,, Doubleday, 1950, Garden City, N,Y,---

6, Goetz, R,C,, "Manipulator systems developed at ANL", Proc, 12th Conf on Remote Systems Technology, ANS, Nov 1964, pp,117-136,

7, ----------, "Fundamentals of general purpose remote manipulators", Nucleonics, 10:11, Nov, 1952, pp, 36-42,

8, ----------, "Manipulators used for handling radioactive materials", Chap 27 of Human Factors in Technology, McGraw-Hill, 1963, New York,

9, Mosher, R,S,, "Industrial Manipulators", Scientific American, Oct, 1964, pp, 88-96,

10, AMF Atomics, Greenwich Conn,, "Model 8 Manipulator", (Manufacturer's Brochure)

11, Central Research Laboratories, Inc,, Red Wing, Minnesota, "CRL Master Slave Manipulators", (Manufacture's Brochure)

12,Hunley, W,H,, Houch, W,G,, "Existing underwater manipulators", American Soc, Mech, Eng, paper no,

13. Karinen, R.S., "Land-based remote handling background of underwater handling equiptment", ASME pap.65-UNT-7

14. Ernst, H.A., A computer-operated mechanical hand", Sc.D Thesis, MIT, Dept. E.E. Dec. 1961.

15. Project MAC, Artificial Intelligence Project Memo Series, MIT.

16. Pingle,K.K., Singer,J.A., Wichman,W.M., "Computer control of a mechanical arm through visual input", IFIP 1968 Congress, Edinburough,

17. Pieper, D.L., "The kinematics of manipulators under computer control", Memo no. A.I. 72, Stanford University,1968

18. Wichman, W.M., "Use of optical feedback in the computer control of an arm", Memo no. A.I. 55, Stanford University, 1967.

19. Reddy, D.R., "Computer Recognition of connected speech", J Acoust Soc Am 42 329-347, 1967

20. Reddy, D.R., Vincens, P.J., "Spoken message recognition by a computer",A.I. Prog. Memo , Stanford University, 1968.

21. Kahn, M.E., "The near-minimum-time control of open-loop articulated kinematic chains", Memo no. AIM-92, Stanford University, 1969

22. Scheinman, V.D., "Design of a computer controlled manipulator", Memo no. AIM-92, Stanford University, 1969.

23. Kay, H.W., "Analysis and discussion of nine externally-powered prosthelc hands", Memo, Comittee for Prosthetic Research and Development, National Research Council.

24. Capowski, J. Personal Communication

25. Englebart, D.C., English, W.K., "A research center for augmenting human intellect", Proc. AFIPS FJCC 1968, pp. 395-410.

26. Sproull, R.F., Sutherland, I.E., "A clipping divider", Proc. AFIPS FJCC 1968, pp.765-775.

27. Sutherland, I.E., "A head mounted three dimensional display", Proc. AFIPS FJCC 1968, pp. 757-764.

28. Roberts, L.G., "The Lincoln wand", Proc. AFIPS FJCC 1966, pp.223-228.

29. Mosher, R.S., "Handyman to Hardiman", SAE pap. no. 670088, Automotive Engin. Conf., Detroit, Mich., Jan. 1967.

30. Mosher, R.S., "Human control factors in walking machine A.S.M.E. Human Factors, Nov. 1965.

31. Johnsen, E.G. and Corliss, W.R., "Teleoperator control", NASA, SP-5070, Dec. 1968.

32. Smith, L., Conversation

33. Wilson, A.B., Exec. Dir. Committee on Prosthetics Research and Develment, Nat'l Acad. Sci., Personal communication.

34. Will, P.M., Zeldman, M., "A high performance ac position servo using a dc motor", IEEE Trans., Vol IECI-14, no. 2, Dec. 1967, pp. 41-46.

35. Scott, R.N., "Myoelectric control systems", Adv. in Biomedical Engineering and Medical Physics, Vol 2, 1968, John Wiley & Sons, New York.

36. Wirta, R.W., Taylor, D.W., "Developement of a multiple-access myoelectrically controlled prosthetic arm", Presented at 3rd Intl. Symp. on External Control of Human Extremities, Dubrovnik, Yugoslavia, Aug. 1969.

37. Wirta, R.W., Taylor, D.R., Finley, F.R., "Engineering principles in the control of extenal power by myoelectric signals", Arch. of Phys. Med. Vol 49:295-296, May 1968.

38. Finley, F.R., Wirta, R.W., Cody, K.A., "Muscle synergies in motor response", Arch. of Phys. Med. Vol 49:11, pp. 665-670, Nov. 1968.

39. Unimation, Inc. Danbury, Conn., "Unimate", (Manufacturer's brochure).

40. Simon, R.C., "Versatran industrial robot applications", A.S. of Tool and Manufacturing Eng. pap. no. AD67-162, Creative Manufacturing Seminars, 1967.

41. Kearney and Trecher Corp., Milwaukee,
"The TURN-12 machining center",
(Manufacturer's brochure),

42. Sutherland, I.E.,
"The ultimate display", Proc. IFIPS
1965, pp. 506-508,

43. Feldman, J.A., Feldman, G.M., Falk, G.,
Grape, G., Pearlman, J., Sobel, I.,
Tenenbaum, J.M.,
"The Stanford hand-eye project",
Proc. Intl. JCC on A.I. May 1969.

44. Sheridan, T.B., Ferrell, W.R.,
"Human control of remote computer-manipulators", proc. Intl. JCC on
A.I., May 1969, pp. 483-494.

PROCESSING MEMORIES

J. P. Bartlett, Fairchild Research and Development Laboratory

ABSTRACT

Various partitioning methods for digital systems are examined to determine what logical functions (special logic consisting of individual gates, registers, bit slices, or memories) are best suited for LSI to yield the maximum logic per dollar. Extrapolating from this information, building blocks are derived with simple cell structures composed into X-Y arrays which look like memories (i. e., processing memories) but can operate as direct replacements for digital logic: registers turn into multiport memories; priority encoders into queue resolving arrays, etc. These and other memories are discussed in detail.

INTRODUCTION

Various partitioning methods for digital systems are examined under the constraint of pin count and defect density to determine what products are useful for Larger Scale Integration (LSI). Simple cell structures composed into X-Y arrays which look like memories (i. e., processing memories), are discussed as possible replacements for digital logic.

LSI Considerations

If LSI is defined as highly complex chips with gate counts of 500 to 5000 gates per device, we can ask is LSI useful; what functions should be made in LSI, and what are the requirements for these functions?

The requirements that LSI must meet are relatively straight forward:

1. LSI must yield the lowest cost solution to the system problem.

   To be of most interest, LSI must be the lowest cost solution in a commercial computer-type of environment; not just ultra high reliability, small size, minimum weight applications.

2. To achieve low cost leads to the next requirement for LSI which is high volume to reduce the cost of manufacturing. Traditionally, large cost reductions in semiconductors came about with large volume. Therefore, LSI products must fit many system requirements, large and small.

3. The effect of package pin count must be considered in the total economic picture.

   In addition, the system cost of mounting the packages and interconnecting them is very nearly proportional to package pins. Most system manufacturers could readily determine figures of cost per pin for any given system fabrication approach they use. Therefore pins should be minimized.

4. It is highly desirable that the total number of different products remain small.

   One can see in today's MSI (Medium Scale Integration) products a 4-bit register with affirmative and negation outputs; 4-bit registers with only affirmative but parallel load capability; 4-bit registers with only affirmative outputs but with outputs driving strobe gates for bussing, etc. All of these products are trivially different. A pattern must be set on which type of products are necessary and how new ones will be generated. The total number must be small.

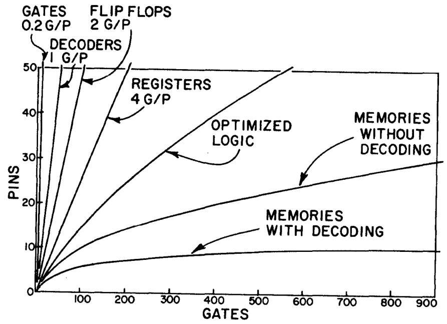
5. LSI products must be economically testable; the arrays need to be tested.

   Testers need to exist to test the devices; programs need to be generated in a reasonable amount of time, and the total cost should be reasonable. Arrays with buried states can require extremely long sequences for functional test.

LSI Partitioning

Can LSI meet these constraints? Looking first at SSI (Small Scale Integration), semiconductor manufacturers first produced single gates and single flip flops on a chip. These gave a very poor gate to pin ratio, i. e., 0. 2 gates per pin for gates and 2 gates per pin for flip flops.

This type of partitioning, while universal, at today's yields would give a product with all the cost in the package and not in the chip. To circumvent this in MSI, single functions were put into a package and higher gate to pin ratios resulted: for decoder and combinatorial networks, a gate to pin ratio of 1 which compares with . 2 for gates in SSI; for registers and counters a ratio of 4 compared to 2. However, both of these partitioning methods are essentially linear in pin count and will not suffice for LSI. Looking at Figure 1, we see memories with and without on-chip decoding of the address give very high gate to pin ratios. With on-chip address decoding, it is possible to double the complexity of the chip with each additional pin. One can generate the curve for memories without on-chip decoding by keeping the input pins equal to the output pins. Optimized logic gives a high gate to pin ratio but is not definable into products. Optimized logic is somewhere in between memories and MSI in gate to pin ratios.



GATES TO PIN CURVES

Figure 1

If the assumption is made that all gates are equal in area, it is possible to examine yield information and see what effect a reduction of 5 in the defect density of the silicon represents for registers and for memories without address decoding.

Calculating the yield, using the formula derived by Dr. Seeds(1), relating chip defect density and chip area to yield, it is possible to generate Table I. Assuming today's defect density of 2.3×10^{-4} per square mil, and a defect density improved by a factor of 5 to give $.46 \times 10^{-4}$ per square mil, the chip yield for registers improves by 1.39 for a 10-pin package and by 3.38 for a 50-pin package. The improvement for registers is small but memories are another story. Here, the yield improves by a factor of 2.00 at 10 pins and 14.35 at 30 pins. These figures are based on 50 square mil per gate with no overhead for metal interconnections. Normally, this assumption on overhead is not valid for random logic gates.

$D_0 = 5D_0'$

YIELD IMPROVEMENT = $\dfrac{\text{YIELD AS A FUNCTION OF } D_0'}{\text{YIELD AS A FUNCTION OF } D_0}$

GATE AREA = 50 SQUARE MILS

YIELD IMPROVEMENT		
PINS	MEMORIES	REGISTERS
10	2.05	1.39
20	6.37	1.82
30	14.35	2.30
40		2.80
50		3.38

YIELD IMPROVEMENT

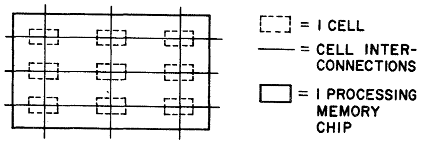
Table 1

The cost ratio of a semiconductor package and the chip it contains remains relatively constant for optimum cost. That is as the chip yield improves from an economic standpoint, economic pressures tend to increase the size and complexity of the chip to maintain this constant.

Clearly then, if semiconductor manufacturers are going to reduce the defect density and thereby improve yields, it is necessary to look at memories or memory-like structures to achieve the promise of LSI.

PROCESSING MEMORIES

Processing memories are defined as memories having the ability to perform logic manipulation on their internally stored data and act as direct replacement for current digital logic. Physically they are represented in Figure 2 as small cells repeated many times on a chip.



A PROCESSING MEMORY

Figure 2

What are the requirements of memory structures for logic? First of all, processing memories should be composed of simple cell functions because design, testing, and understanding of the cells then becomes relatively straight forward. It is possible for the human to design and optimize a simple cell to achieve the most function per dollar but a large chip of random gates is impossible to optimize because of the number of variables involved. Testing is becoming another major cost factor. Currently, minutes are spent testing one device and days are spent generating the test procedure even with a high degree of automation. The generation of algorithms for testing simple cells should be relatively straight forward.

Processing memories are easy for the customer to understand. Education is expensive and a measure of this is the time it takes an engineer to understand all of the functions and operating procedures for a logic function composed of 3000 random gates.

Secondly, to get complex modules, repetitive uses of these simple cells must form useful functions. Since the cells are repetitive, one makes an array of cells in an area which is compatible with current yield requirements. This way complex modules are easily made, good yields obtained and the cells are easy to interconnect. Cell metal interconnection tolerances are the worst tolerances in semiconductor technology and difficult interconnection problems waste silicon area.

Finally, these arrays must be easily expandable by words and by the number of bits per word. This allows the arrays to "fit" both small and large digital system requirements. In addition, as yield improves, larger arrays are made out of the same simple cells. This gives the user product trends, he knows what new products are coming and he can incorporate them into his system easily without major redesign. Re-education is nil.

To find what types of products to make as a starter set of products, it is possible to look at the history in integrated circuits. In examining the changes in going from SSI to MSI, one finds SSI put single gates or a single flip flop on a chip. Then because of pin problems, MSI interconnected gates and flip flops in the package to make single functions. These looked like horizontal or word partitioning of digital systems. All of the MSI functions are nonprogrammable with the exception of a few instances, i. e. , a 4-bit counter might have a programming pin to allow making the counter divide by 16 or by 10.

The easiest extension for new products in MSI was going from a flip flop in SSI to 4-bit registers. AND gates in SSI to be useful and reduce pin counter turned into decoders, (i. e. , BCD to decimal, BCD to 7 segment, etc.) As functions became more complex, the mixing of the AND's and OR's function on a single chip was possible giving selectors and multiplexers. At this time definition of new products becomes very difficult (hence multiplicity of same products.)

To expand into LSI, the first step is to go to multiple functions per chip. Instead of one encoder per chip, make many encoders per chip, i. e. , go to multiple input/multiple output functions. One obtains the most functions per dollar when the output pins equal the input pins unless on-chip decoding is included; that is, combine horizontal and word partitioning into the same array. By making the same function, say encoders many times on the chip, one introduces the idea of repetitive cells into the chip design. Finally, by making the chip programmable, it is possible to meet many different encoding needs. Following these rules, it is easy to extend MSI to LSI.

First, look at decoders, they are just a group of AND gates which detect a code on the input. An associative memory does the same thing when it generates a match indicating the input equals the stored data, it is nothing more than AND gates. Encoders are just random access memories. One energizes the select line and the output represents the code for that select line. Registers turn into multiport memories and in this case, they also include the selection logic for the registers. These are obvious ones but others are definable. Selectors and multiplexers are just routing networks which turn into interconnect arrays (2) which can route any input to any output simultaneously under program control. Parallel to serial converters and serial to parallel converters can be incorporated into a corner-turning buffer. (A corner-turning buffer loads one word at a time and reads out one bit of every word at a time.) Priority encoders are essentially sorting arrays which turn into queue resolving arrays. Decoders to handle control functions turn into control associative memories. These are capable of taking advantage of logic minimization by forcing a match at any bit location in the memory. Table II summarizes these results. It is possible to define a minimum set of memories from which it is possible to build most digital systems.

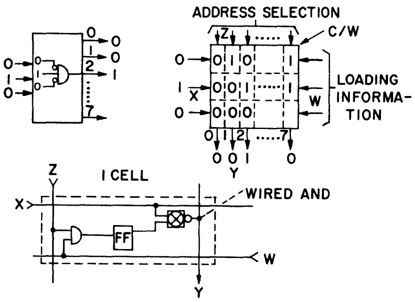
SSI →	MSI →	LSI
AND GATES	DECODERS	ASSOCIATIVE MEMORIES
OR GATES	ENCODERS	RANDOM ACCESS MEMORIES
FLIP FLOPS	REGISTERS	MULTIPORT MEMORIES
AND/OR GATES	SELECTORS MULTIPLEXERS	INTERCONNECT ARRAYS
	PRIORITY ENCODERS	QUEUE RESOLVING ARRAYS
	PARALLEL TO SERIAL CONVERTERS	CORNER TURNING BUFFERS
	FUNCTION CONVERTERS	CONTROL ASSOCIATIVE MEMORIES

LOGIC COMPLEXITY TRENDS

Table II

Associative Memory

Figure 3 shows how a decoder turns into an associative memory. The decoder is a bank of AND gates set up to recognize various codes on the input. An associative memory can perform the same function. The words of the associative memory are loaded with ones for detecting the affirmative and with zeros for detecting the negation, i. e., in the decoder in Figure 3, the AND gate is detecting the code 010 and putting out a response for number two output. In the associative memory the same code is entered into the memory and word two generates a match signal saying that word two is equal to the input word. The only added complexity of the associative memory over the decoder is the ability to store information and the necessity to load this information before using it. At the bottom of Figure 3 is an illustration of the cell. The "X" represents an input line which is compared with the contents of the flip flop and the output of the detector is AND'd with all the other bits in that word. If all the flip flops match the inputs, then "Y" indicates a match condition. "W" is the input data during loading and "Z" is the address line to select that flip flop at loading time. Applications are code conversion. One can easily expand the cell capability by putting in a bit masking capability which allows associative processing of data.

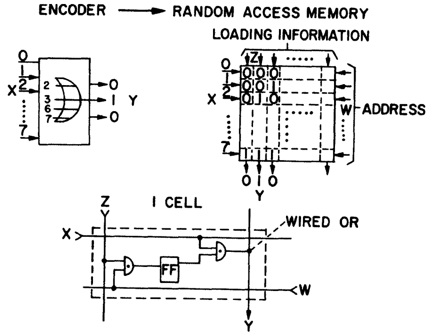


AN ARRAY OF AND GATES

Figure 3

Random Access Memory

Encoders are easier than decoders. Figure 4 has an OR gate illustrating an encoder where the output "Y" is energized wherever inputs 2, 3, 6, or 7 are enabled. To see the logical OR function or encoding function in Random Access Memory (RAM), only consider one bit in each word of the RAM, say Column Z. Energize the linear select line, say "X" to the RAM and the output, say "Y", is either a one or a zero depending on what is stored in the memory, i. e., store a one in the bit positions on select lines 2, 3, 6, and 7, and zero on bits positions 0, 1, 4, and 5, then output "Y" of the RAM corresponds to output "Y" of the encoder. So expanding the number of bits per word turns the random access memory into a group of OR gates. Figure 4 shows the cell as only a flip flop and two gates. Uses are decimal to binary conversion, etc.



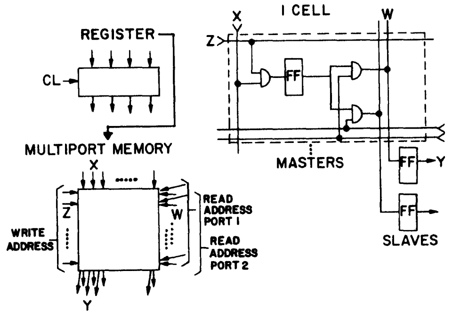
AN ARRAY OF OR GATES

Figure 4

Multiport Memories

The extention of registers to multiport memories requires a little more than the first two arrays because it is necessary to include the associated selection logic for the registers. In digital systems, this is usually the bus structure on the registers or the selecting logic in radially structured systems. The Multiport Memory (MPM) still maintains the master slave flip flops associated with registers; however, to supply complete accessibility to every register simultaneously is prohibitive in pin count. One builds the MPM by building ports for accessibility to the memory. Therefore, it is possible to build MPM with one write port and one read port; one write port and two read ports, two write ports and one read, or some other combination. Looking then at Figure 5, one sees a multiport memory with one write port. "X" is the data and "Z" is the write address location for storing data into the master flip flop (FF).

Coming out of FF are two read ports and their associated selection lines. Each read port feeds into another FF which acts as the slave for the master slave flip flop.



AN ARRAY OF STORAGE ELEMENTS

Figure 5

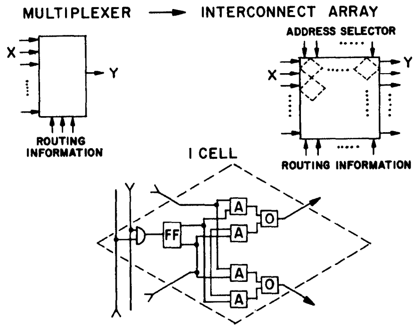
One can easily add features to the device which only increases the complexity slightly. Looking at the read ports where they are shown going to the slave flip flop, it is possible to place a multiplexer on each port. The two read ports now go to four slaves instead of two. This yields a multiport memory with one write port and four read ports. The significance here is the output on only two read ports can change at one time. To summarize, MPM design needs to consider the accessibility of the memory (i. e. , how many ports and what type) and the rate the data is changing through the ports. Finally, to increase the usefulness of the memory, a mask writing function is included on the input write port.

Figure 5 shows the basic cell as only a flip flop and three gates.

With these three blocks, it is possible to build most digital systems (i. e. , ANDs, ORs and storage -- decoders, encoders, and registers -- associative memories, random access memories, and multiport memories), but other useful memory structures are definable. Basically arrays for mixing data follows the rules for simple repetitive cell functions.

Interconnect Array

This is an array which incorporates many selectors or multiplexers into one array. First, looking at Figure 6, any of the inputs are programmable to "Y", say "X" is programmed to "Y". After which all inputs except "X" are programmable to any output except "Y". In other words, any permutation of inputs to outputs is possible. The cells in this memory are designed to act like a switch. Information arrives at two sides of the "diamond" cell and is routed straight through or if the internal flip flop of the cell is set, the information turns by 90 degrees before being routed out of the cell. This cell is composed of a flip flop, five AND gates and two OR gates.



AN ARRAY OF MULTIPLEXER

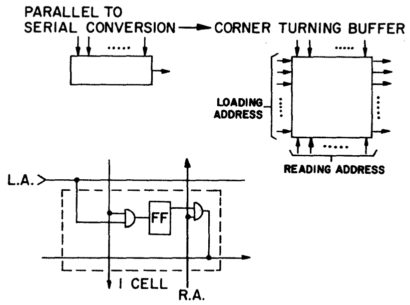
Figure 6

This kind of array is useful for telephone type operations, dynamically alterable computers using variable field format, changing the computer structure during malfunction to increase reliability, and resource allocation problems.

Because of problems in determining whether OR, AND, exclusive OR, etc. , is preferable, the array does not allow multiple inputs connected to a single output. But one should include control on each output pin for enabling/disabling so that the OR tie capability can be realized .

Corner-Turning Buffers

Parallel stream to a serial data stream is a logical form capable of mixing data. Once again, a memory organization will meet this requirement. If information is loaded parallel by word and is read parallel by bit, i. e. , one bit of every word, the memory performs like many parallel to serial converters. This is nothing more than a random access memory with the read selection logic treating each bit of every word as a word for reading. Figure 7 shows the information going in the top of the corner-turning buffer (CTB) and coming out of the side of the CTB. Putting two of the arrays together, it is possible to mix and then unmix the data streams.


PARALLEL TO SERIAL CONVERSION
CORNER TURNING BUFFER
LOADING ADDRESS
READING ADDRESS
L.A.
FF
1 CELL
R.A.

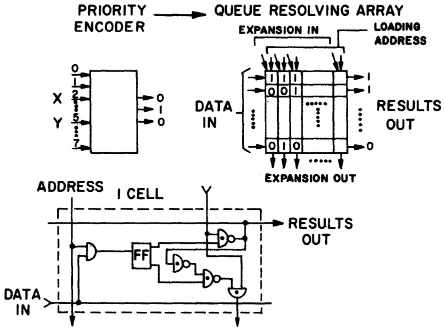
AN ARRAY OF PARALLEL TO SERIAL CONVERTERS

Figure 7

Uses for a CTB are in noisy data channels subject to burst error noise. Including error correction on every word and then mixing burst noise will only effect one bit of the word which can be corrected.

Queue Resolving Array

Ordering data into a sequence is another logical form of mixing data, e. g. , given a list of data which is significant in a binary code, Figure 8 shows the type of cell to sort this information into the lowest number first. In other words, a priority encoder turns into a queue resolving array (QRA). The cell structure is again only a flip flop and a few gates which are repetitive. Information is stored in the memory on one set of lines and the output is always the lowest number currently in the memory.


PRIORITY ENCODER
QUEUE RESOLVING ARRAY
EXPANSION IN
LOADING ADDRESS
X
Y
DATA IN
RESULTS OUT
EXPANSION OUT
ADDRESS
1 CELL
Y
FF
RESULTS OUT
DATA IN

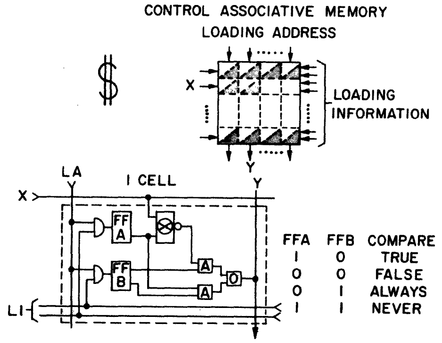
AN ARRAY OF PRIORITY ENCODERS

Figure 8

A QRA memory is useful for priority interrupt systems in computers, sorting numbers, and scheduling algorithms in time sharing and multiprogrammed computers.

Control Associative Memory

Decoders perform the function of AND gates but they have one problem; that is AND products are not reduceable to minimum form. To meet this requirement, a control associative memory (CAM) is useful which allows a programmed forced match at each bit in every word. Figure 9 shows the memory structure. It is composed of two flip flop per cell and the associated equivalents gate with programmable override. Figure 9 explains the coding on the two flip flops for the four compare conditions.


CONTROL ASSOCIATIVE MEMORY
LOADING ADDRESS
X
LOADING INFORMATION
Y
LA
1 CELL
Y
X
FF A
FF B
A
A
O
LI
FFA FFB COMPARE
1 0 TRUE
0 0 FALSE
0 1 ALWAYS
1 1 NEVER

AN ARRAY OF LOGIC REDUCEABLE GATES

Figure 9

If one has a function which reduces down to a few terms, it is built with the CAM. If it does not reduce down but has mutually exclusive gates, the normal associative memory is the one to use. An illustration of this is the function made up of $ABD + A\bar{B}C + \bar{A}B\bar{D}$. This requires six words of associative memory but only three words of CAM.

SUMMARY

Some of the applications for associative memories and random access memories seem to be translation. However, a word of caution is needed as all of these processing memories were discussed without any encoding or decoding internally on the chip or time sharing of input/output pins. Including these features tended to cloud the discussion with many side effects. Translation in the general sense is really AND/OR logic and the associative memory only has AND capability.

Examining the requirement for LSI to succeed shows processing memories can be used like current SSI and MSI, so large volume is possible. The total number of types is reasonably small; seven were defined in this paper but that certainly is not all.

What is the cost?

It looks expensive to build processing memories because one is adding a lot of extra functions to the package which are unnecessary. Table III summarizes the number of square mils per gate for memory devices versus the years they were developed. From 1967 to 1969, the gate size has dropped by a factor of better than five and consequently the gate complexity in the chip has gone from 64 gates to 1100 gates on chip. These products are all memories. Table IV compares a memory chip with a random logic chip in the year 1968. The memory chip has a gate complexity of 2. 5 times that of the random logic chip. If one examines the random logic chip carefully you soon discover there 40% of the chip is used mainly for cell interconnections. Therefore processing memories in terms of silicon real estate and cost are not expensive because the cells are more compact and the interconnection overhead is minimal.

YEAR	PART NO.	GATES/CHIP	CHIP SIZE	AREA/GATE
1967	9033	64	72x68	76.5
1968	9035	256	137x137	74.4
1969	4100	1100	110x140	14.0
1968	R&D ARRAY	1000	70x70	4.9

SILICON AREA OF A MEMORY GATE

Table III

YEAR	PART NO.	GATES/CHIP	CHIP SIZE	AREA/GATE
1968	9035	256	137x137	74
1968	4700	96	145x145	222

SILICON AREA FOR A MEMORY GATE
AND A LOGIC GATE

Table IV

Table V looks at five designs to see how efficient processing memories are for realizing logic functions. The percent efficiency in terms of logic gates is the ratio of raw logic gates to processing memory gates which vary from 8% on random logic functions to 25% for a translator. A central processing unit for a mini computer is 9%. Reason for the lower number with the computer is the random control logic heavily weights the statistic. However, looking at 1970 technology, the ratio of a processing memory gate area to the raw logic gate area is approximately 10. Defining efficiency in terms of silicon area, and hence cost, for the five designs shows that using raw logic gates for random logic is 0. 8 times and for translators is 2. 5 times the cost of a processing memory implementation.

EFFECTIVE GATE COUNT FOR
PROCESSING MEMORIES

LOGIC DESIGN	EFFICIENCY (GATES)	EFFICIENCY (AREA)
I OUT OF 32 DECODER	20%	200%
RANDOM LOGIC FUNCTION	8%	80%
ADC LOGIC	20%	200%
CPU MINI COMPUTER	9%	90%
TRANSLATOR	25%	250%

EFFECTIVE GATE COUNT FOR
PROCESSING MEMORIES

Table V

The number of pins must remain small. In the random access memory instead of using linear select addressing, one could use a binary address. This requires the associative memory to encode the location of the match when using the associative memory with the random access memory for AND/OR functions. It costs time, however, for the encoding and then decoding on each chip and its only function is to save pins. Another approach is to use computer aided design (CAD) and store these cell structures in a library and make the arrays a mixture of cells as per the customer requirements. This is a very easy placement and wire routing algorithm for CAD. By using processing memories with on-chip decoding or mixing cells using CAD, gate to pin ratios become very high.

Testing is simplified since the functions are repetitive and one only needs to specify repetitive patterns to the device under test. Trying to determine logic patterns for complex random logic is extremely difficult and expensive at this level of integration. A test case of a 200-gate function with five buried states took 70 hours on a 360/40 and then only tested 70% of the chip.

CONCLUSION

Currently, the technology is at the 2000 gate processing memory capability which represents a useful gate complexity of 160 to 400 gates per chip.

To produce these designs requires nothing more than memory products announced or in development today. LSI is definable into products and is finally here.

REFERENCES

(1) Internal Communication

(2) IEEE TRANSACTIONS ON COMPUTERS,
VOL. C017, NO. 5, MAY 1968
"Cellular Interconnection Arrays", William H. Kautz, Karl N. Levitt, and Abraham Waksman

THE USE OF READ ONLY STORAGE MODULES TO PERFORM COMPLEX LOGIC FUNCTIONS

Joel C. Leininger, IBM Components Division, EastFishkill Facility, Hopewell Junction, New York 12533

ABSTRACT

A Read Only Storage Module may be used to form large Control Store Arrays or to perform Logic Functions. When used for logic, it may be considered as a functional circuit where the outputs are any possible function of the input variables. In this manner, logic units such as code translators and checkers, shifters, parity checkers, majority logic, two, three, four or five input adders, and other functions may be formed simply from a small number of Read Only Storage Modules.

The overall power of the Read Only Storage Module to perform logic functions opens new vistas in design. Techniques which previously could not even be considered become simple to define and implement. The designer's imagination and the size of the array are the only limits for what can be accomplished.

DISCUSSION

The use of Read Only Storage (ROS) Modules to form large control stores is well known. Read Only Storage Modules can, however, be used to perform complex logic functions. The techniques required are relatively simple with imaginative design being the important key. From a designer's point of view, the organization and composition of the arrays are of secondary importance to the effective and challenging uses of such arrays.

A Read Only Storage operates in a manner similar to a conventional Read-Write Storage. When a word is addressed the data contained in the word is read out. The difference between and Read-Write Storage and a Read Only Storage is that the data in a ROS is determined during manufacture and not readily changeable.

The ROS Module may be considered a "Black Box" where each address line is an input variable and where the output functions are formed by the array outputs. Any logical function of the input variables may be generated, and only one phase of an input variable is required. In examples in this paper, arrays of 256 words with widths of 2 or 4 bits will be used. They correspond to "Black Boxes" with 8 inputs and 2 or 4 outputs. Each array output may be represented by an 8 variable Karnaugh Map where the output is the function represented by the map. A map with a single "one" in it corresponds to an 8 way And.

For a better indication of the relationship between inputs and outputs see Fig. 1.

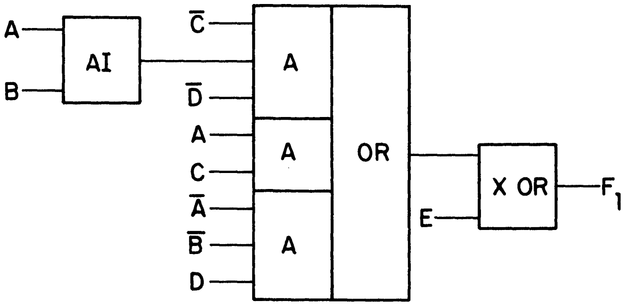


Fig. 1. Circuit layout for a 5 variable function.

The Karnaugh Map and Truth Table for such a function are given in Fig. 2 and Table I.

CD \ AB	00	01	11	10	00	01	11	10
00	1	1		1			1	
01	1					1	1	1
11	1		1	1		1		
10			1	1	1	1		
		$\overline{E}$				E		

Fig. 2. Karnaugh Map.

Table I. Truth Table for a 5 variable function.

A	B	C	D	E	F_1
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	0	0	1
1	1	1	1	0	1
1	1	1	0	0	1
1	0	0	0	0	1
1	0	1	1	0	1
1	0	1	0	0	1
0	0	1	0	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	0	0	1	1	1

This 5 variable function may be obtained by using a ROS Module containing a five input, one output array. The ones in the above Karnaugh Map and Truth Table would represent those built into the ROS array to provide function F_1.

To apply ROS Modules to logic, one simple approach is to take existing logic and look for areas where the proper number of inputs and outputs are present. This is not a very imaginative approach and does not usually result in a significant logic reduction. Consider a typical adder. When looking at individual logic pages there are many lines entering or leaving. The use of ROS Modules in this case is not at all obvious.

A much better approach for the application of ROS Modules is to look at the overall picture of various functional areas. The overall picture of an adder, for example, is quite simple with a relatively small number of input and output lines. Better still, the use of the ROS Module should be considered when determining the machine organization in the same manner by which circuit speed, fan in, fan out, dotting and driving rules are considered in machine design. A desired machine performance may be achieved with faster circuits, a wider data path, the addition of a shifter, a three or more input adder, or various other trade-offs. Some of these trade-offs are only apparent when the use of the ROS Module is considered in determining the machine organization.

Some of the applications of the ROS Module are adders, majority logic, shifters, code translators and checkers, register detects, and register compare functions. Figures 3a and 3b show a two-input adder and a five-input adder. In the two-input adder, each ROS Module replaces up to 35 circuits and 5 levels of logic. There are many other adders possible including adders with look ahead and adders with the logical functions And, Or, and Exclusive Or included. For the two-input adder shown in Fig. 4 three control lines may be used and decoded to give the functions shown in Table II.

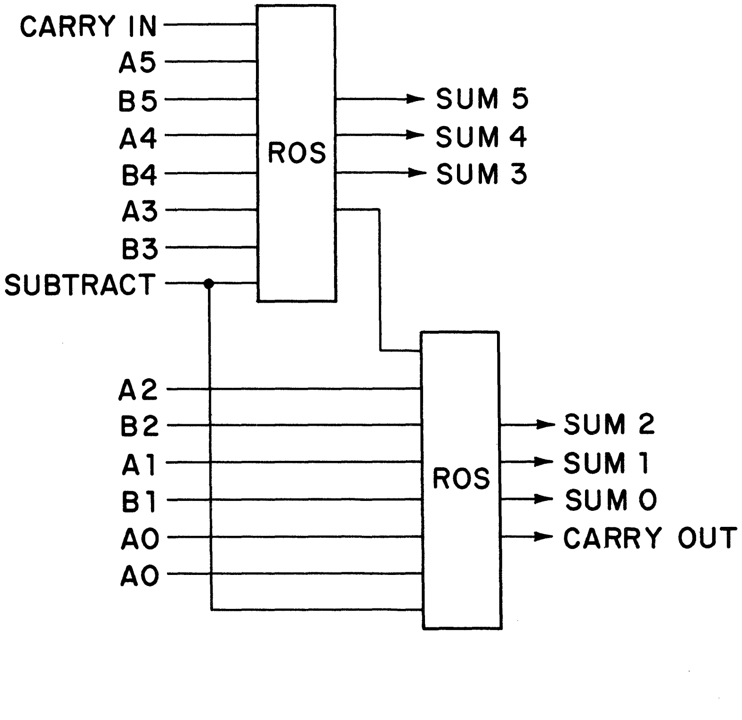


Fig. 3a. Six bit section of 2 input adder.

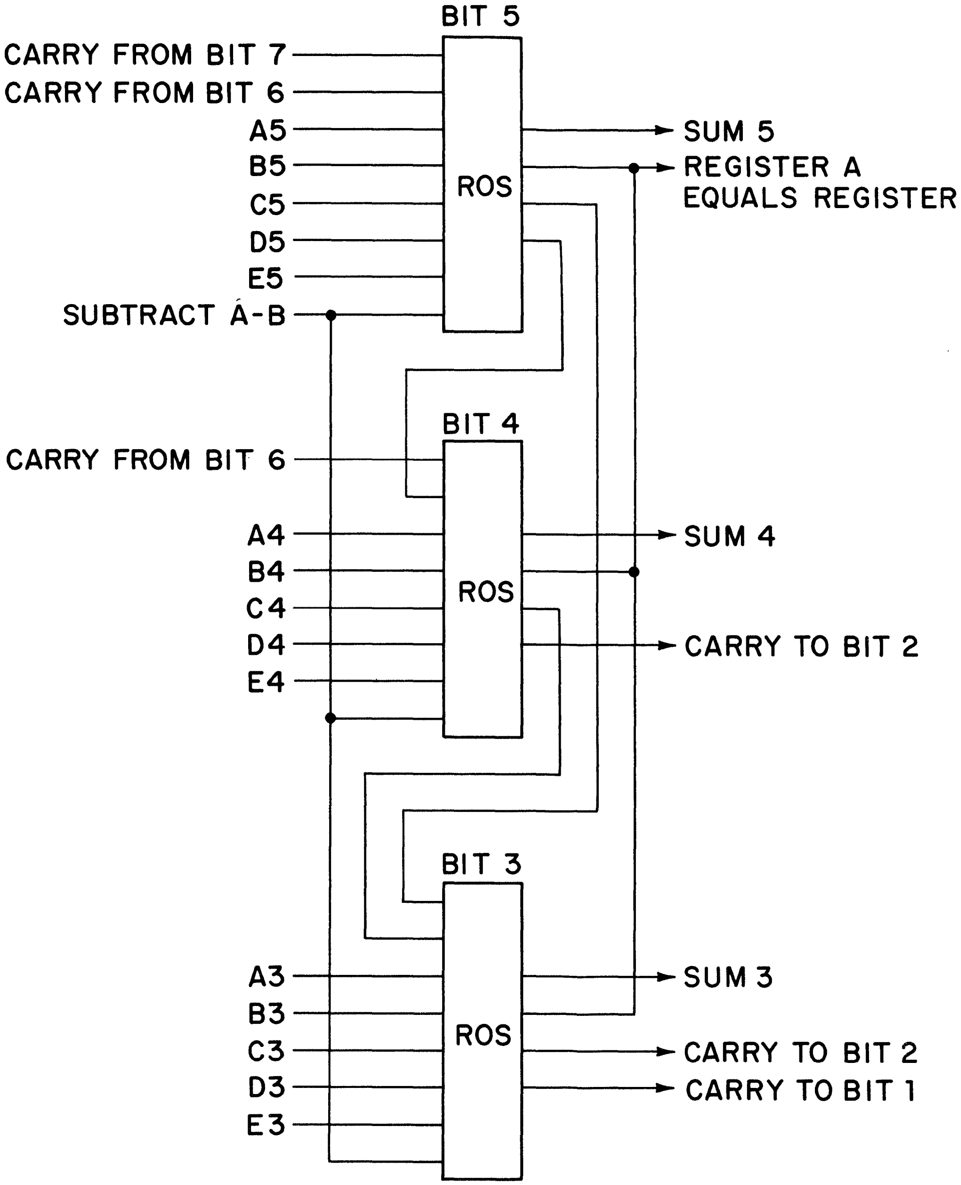

BIT 5
CARRY FROM BIT 7
CARRY FROM BIT 6
A5
B5
C5
D5
E5
SUBTRACT A-B
ROS
SUM 5
REGISTER A EQUALS REGISTER
BIT 4
CARRY FROM BIT 6
A4
B4
C4
D4
E4
ROS
SUM 4
CARRY TO BIT 2
BIT 3
A3
B3
C3
D3
E3
ROS
SUM 3
CARRY TO BIT 2
CARRY TO BIT 1

Fig. 3b. Three bit section of 5 input adder.

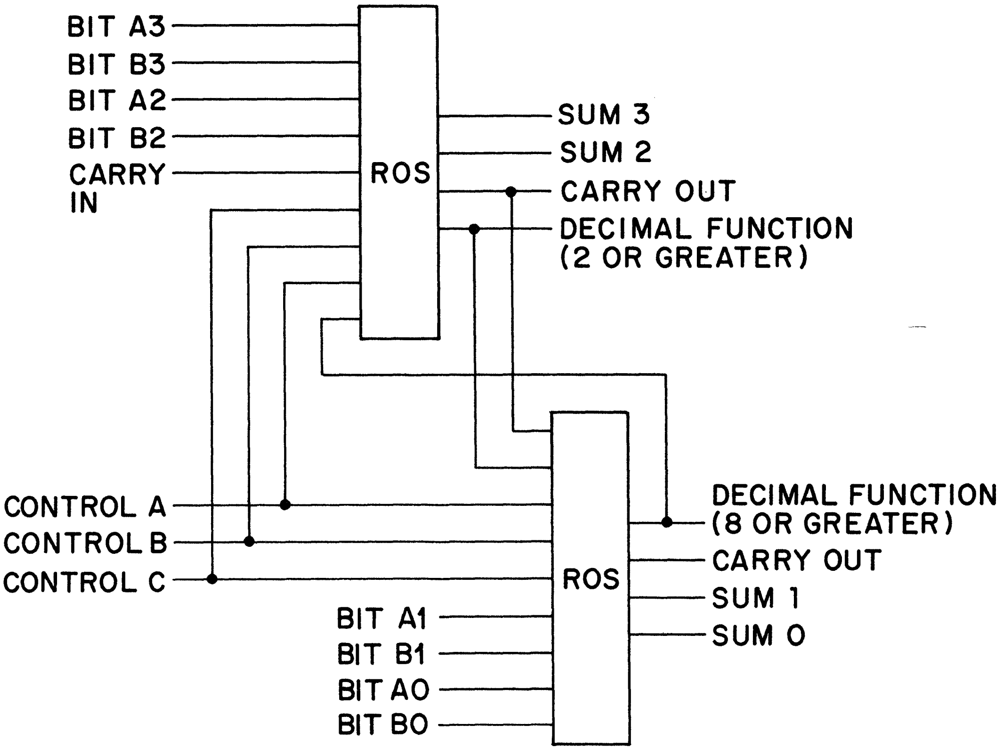

BIT A3
BIT B3
BIT A2
BIT B2
CARRY IN
ROS
SUM 3
SUM 2
CARRY OUT
DECIMAL FUNCTION (2 OR GREATER)
CONTROL A
CONTROL B
CONTROL C
DECIMAL FUNCTION (8 OR GREATER)
CARRY OUT
SUM 1
SUM 0
ROS
BIT A1
BIT B1
BIT A0
BIT B0

Fig. 4. 4 Bits of Binary - Decimal Adder.

TABLE II. Functions for a two-input adder.

Adder Function	Control A	Control B	Control C
Output Zero (or equal input A)	0	0	0
Add	0	0	1
Decimal Add	0	1	0
Subtract	0	1	1
Decimal Subtract	1	0	0
And	1	0	1
Or	1	1	0
Exclusive Or	1	1	1

Majority logic is used in character readers and is quite adaptable to ROS (see Fig. 5). When determining a horizontal line, a vertical line or some other portion of a character, more points are sensed than are required to determine the function. If eight points are sensed, five or more being present may be sufficient to determine that the function is present. To determine the number of functions or combinations of M objects taken N at a time, the equation is $M!/N!\,(M-N)!$

Using conventional logic for the above example, a total of 56 five-way And circuits with the output ored together would be required. Using the ROS Module, however, only one-half or one-fourth of the module would be required to generate the function, and other functions of the inputs could also be generated.

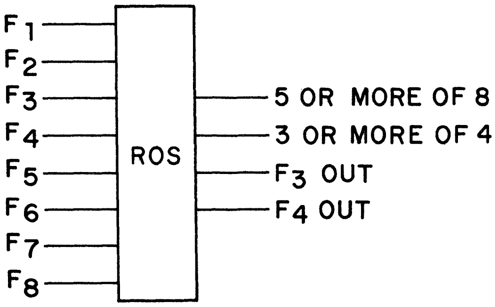

F1
F2
F3
F4
F5
F6
F7
F8
ROS
5 OR MORE OF 8
3 OR MORE OF 4
F3 OUT
F4 OUT

Fig. 5. Majority Logic.

Figure 6 shows uses of the ROS Module in code translation and checking, and in a sixteen bit register detect and compare application. More complex code translations can be handled with more than one level of the ROS Module. In the above type of applications, module reductions of six or more to one can be achieved compared to standard logic modules. In one translation application, four ROS Modules replace 400 logic circuits.

The use of the ROS Module to build a 32 bit shifter is shown in Fig. 7. The shifter will basically shift right but by interchanging the corresponding high order and low order bits into the first and last stages, both left and right shifts can be performed. The shifter contains 33 ROS Modules and will perform both left and right logical shifts from 0 to 31 positions. In addition, the sign may be propagated on arithmetic right shifts.

The preceding examples show just a few of the possible applications of the ROS Module. The ROS Module is a powerful design tool that provides a significant reduction in the number of modules required to perform complex logic functions.

The advantages of using the ROS Module to perform logic functions are numerous. Among the most important are:

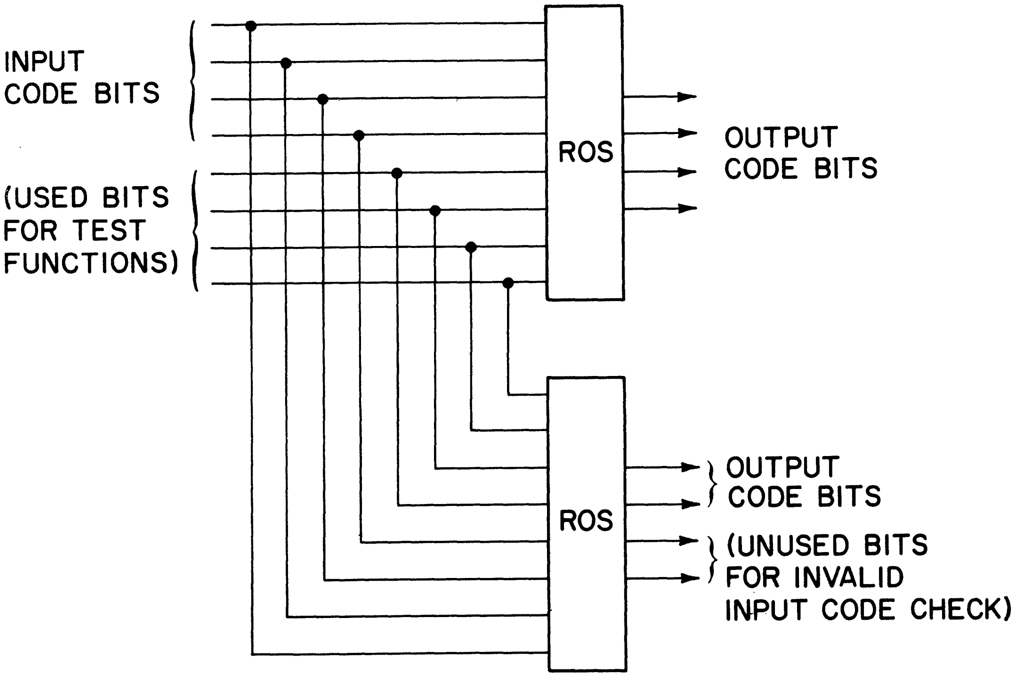
1. A size and cost reduction which will reduce the number of modules used and may result in further savings on cards, boards, cables, power supplies, etc. and provide a smaller overall package.
2. Improved reliability since there are fewer connections, cables contacts, and modules.
3. A quicker diagnosis of failures since there will be fewer replaceable units and more logic on each unit.
4. A savings in field stocking of parts and inventory.
5. A possible increase in speed since the ROS Module may replace many levels of logic.

The major disadvantage of the ROS Module is the turnaround time required to make changes to the output functions.[1] The number of changes required may be significantly reduced to help circumvent this drawback, by simulation of the output functions.

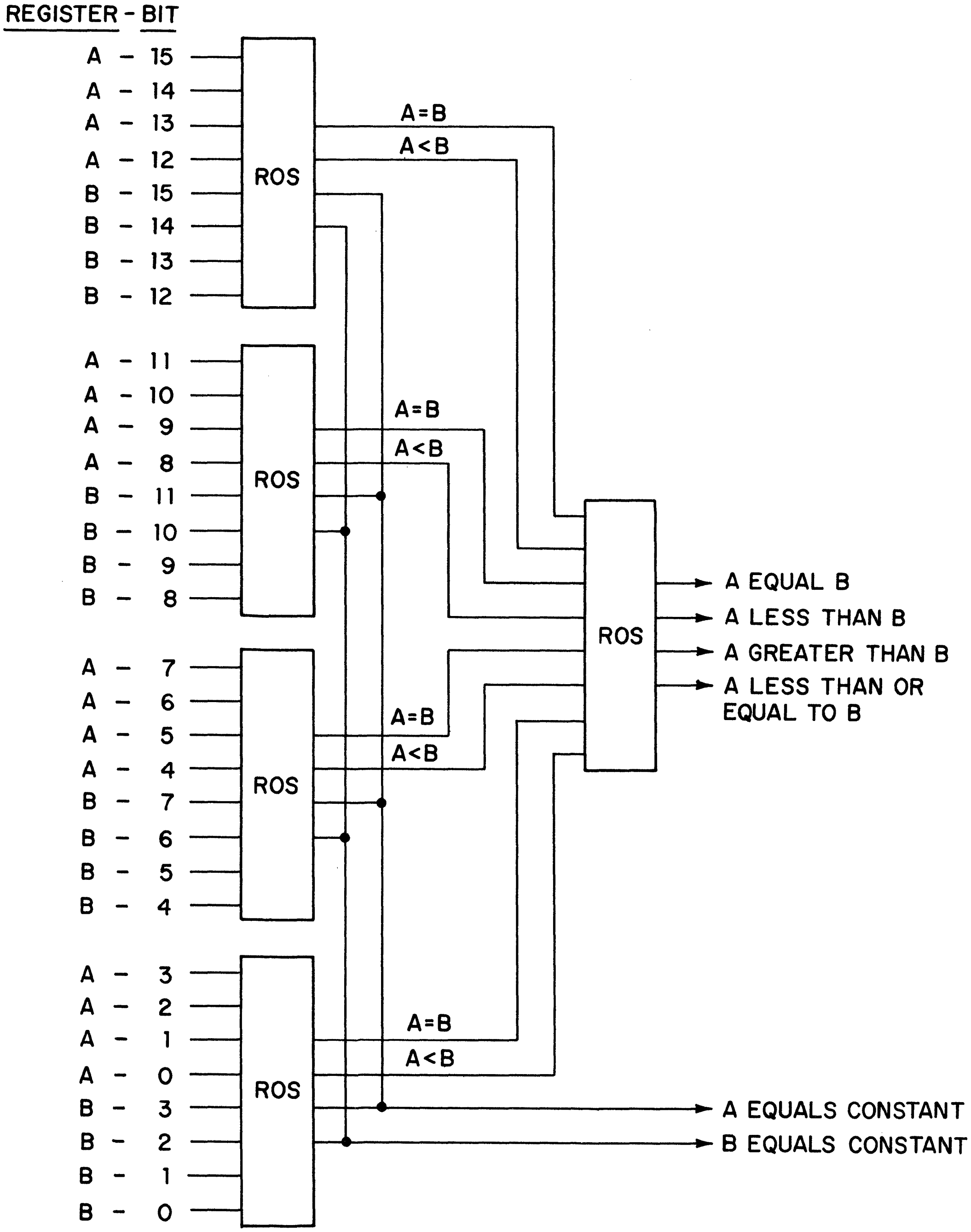
The ROS Module is a new powerful logic design tool. It offers increased circuit density, reduces wiring and packaging problems, and can be used to generate the most complex logic functions. It presents a challenge to designers and implementers in all logic design areas. Not only does it affect the implementation of machine hardware, but it has a significant influence on the organization of machines and makes designs that were previously impossible to consider a reality to build.

REFERENCE

1. Linford, J. "ROM at the Top," Electronic Engineer, May, 1969.


INPUT CODE BITS
(USED BITS FOR TEST FUNCTIONS)
ROS
OUTPUT CODE BITS
ROS
OUTPUT CODE BITS
(UNUSED BITS FOR INVALID INPUT CODE CHECK)

Code Translation and Checking


REGISTER - BIT
A - 15
A - 14
A - 13
A - 12
B - 15
B - 14
B - 13
B - 12
A - 11
A - 10
A - 9
A - 8
B - 11
B - 10
B - 9
B - 8
A - 7
A - 6
A - 5
A - 4
B - 7
B - 6
B - 5
B - 4
A - 3
A - 2
A - 1
A - 0
B - 3
B - 2
B - 1
B - 0
ROS
ROS
ROS
ROS
ROS
A=B
A<B
A=B
A<B
A=B
A<B
A=B
A<B
A EQUAL B
A LESS THAN B
A GREATER THAN B
A LESS THAN OR EQUAL TO B
A EQUALS CONSTANT
B EQUALS CONSTANT

Register Detects and Compare Functions

Fig. 6. Uses of the ROS Module.

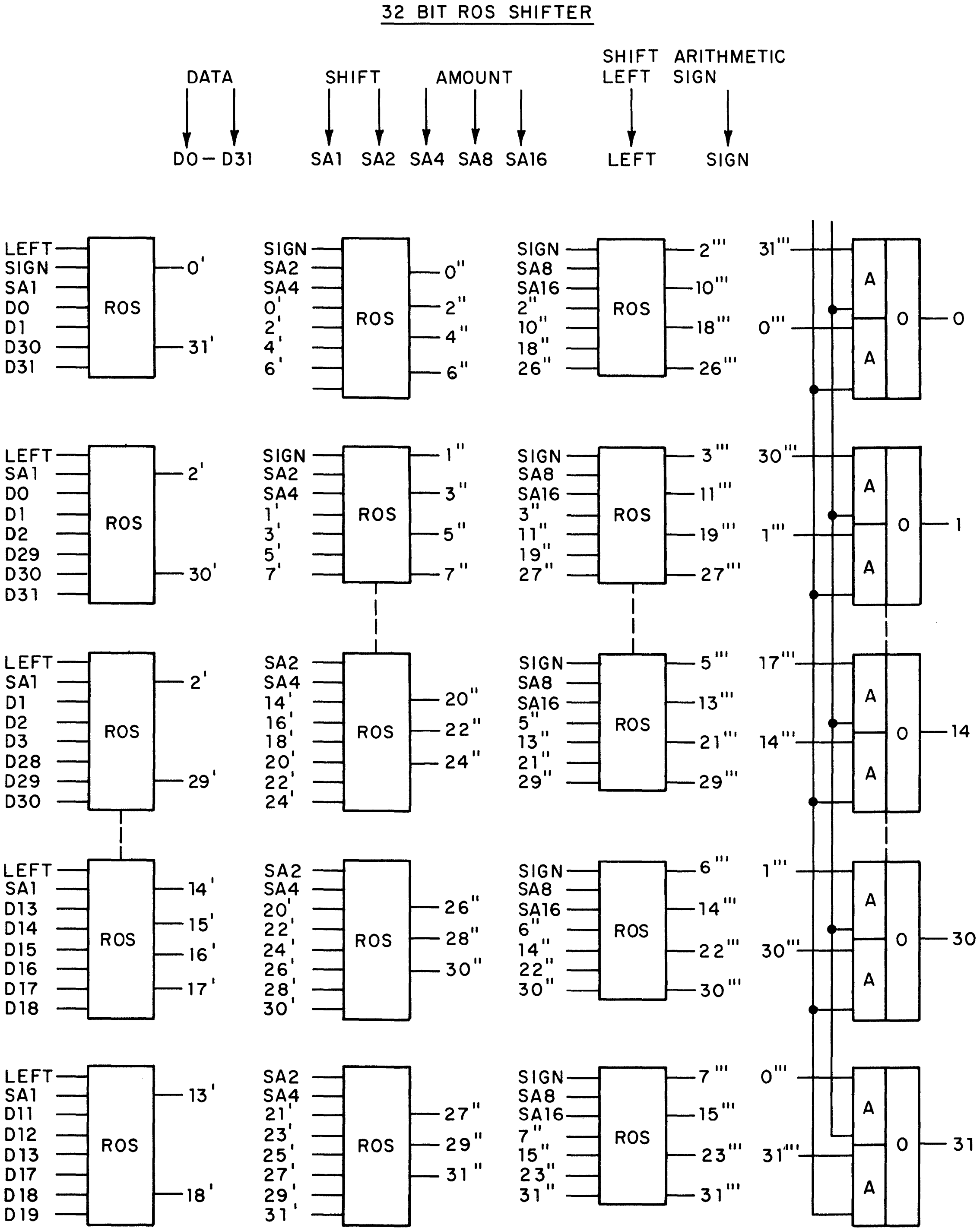

32 BIT ROS SHIFTER
DATA
SHIFT AMOUNT
SHIFT LEFT
ARITHMETIC SIGN
DO – D31
SA1 SA2 SA4 SA8 SA16
LEFT
SIGN
LEFT SIGN SA1 DO D1 D30 D31
ROS
0'
31'
SIGN SA2 SA4 0' 2' 4' 6'
ROS
0"
2"
4"
6"
SIGN SA8 SA16 2" 10" 18" 26"
ROS
2'''
10'''
18'''
26'''
31'''
0'''
A
A
O
0
LEFT SA1 DO D1 D2 D29 D30 D31
ROS
2'
30'
SIGN SA2 SA4 1' 3' 5' 7'
ROS
1"
3"
5"
7"
SIGN SA8 SA16 3" 11" 19" 27"
ROS
3'''
11'''
19'''
27'''
30'''
1'''
A
A
O
1
LEFT SA1 D1 D2 D3 D28 D29 D30
ROS
2'
29'
SA2 SA4 14' 16' 18' 20' 22' 24'
ROS
20"
22"
24"
SIGN SA8 SA16 5" 13" 21" 29"
ROS
5'''
13'''
21'''
29'''
17'''
14'''
A
A
O
14
LEFT SA1 D13 D14 D15 D16 D17 D18
ROS
14'
15'
16'
17'
SA2 SA4 20' 22' 24' 26' 28' 30'
ROS
26"
28"
30"
SIGN SA8 SA16 6" 14" 22" 30"
ROS
6'''
14'''
22'''
30'''
1'''
30'''
A
A
O
30
LEFT SA1 D11 D12 D13 D17 D18 D19
ROS
13'
18'
SA2 SA4 21' 23' 25' 27' 29' 31'
ROS
27"
29"
31"
SIGN SA8 SA16 7" 15" 23" 31"
ROS
7'''
15'''
23'''
31'''
0'''
31'''
A
A
O
31

Fig. 7. Use of the ROS Module to build a 32 bit shifter.

FUNCTIONAL MEMORY AS A GENERAL PURPOSE SYSTEMS TECHNOLOGY

by

M. Flinders, P. L. Gardner, R. J. Llewelyn, J. F. Minshull

IBM United Kingdom Laboratories Limited.

ABSTRACT

Large Scale Integration (LSI) offers the equipment designer a dramatic reduction in circuit cost, at the expense of changeability during development and in the field. The semiconductor manufacturer is faced with an explosion in the number of chip types required and he becomes involved in a much more complex interface with the system designer.

In order to overcome these problems this paper proposes a Functional Memory module, based on a loadable array of three state associative cells, as a general purpose system technology.

Logical operations in Functional Memory are performed using table look-up methods. The paper describes a feasible FM module and the way in which tables for various logical operations can be designed. Modules may be grouped together to form a system data flow, and an example is given.

The use of this technology introduces a number of checking problems not normally encountered in conventional logic. Solutions to these problems have been devised which link effective fault detection and location with a potential for self-repair.

The paper concludes that this type of associative array logic can exploit the potential of LSI while eliminating some of the problems of today's conventional technology.

Functional Memory technology capitalises on the high circuit densities offered by LSI and is more versatile than current logic technologies. It allows functions to be changed easily and avoids the part number explosion.

ACKNOWLEDGEMENTS

The authors wish to acknowledge ideas contributed by numerous colleagues and to thank the management of International Business Machines Corporation for their permission to publish this paper.

INTRODUCTION

The advent of Large Scale Integration (LSI) promises significant reductions in the cost of electronic circuits. However, for the designer of complex digital systems it introduces a range of problems for which existing solutions are, in most cases, inadequate. He is faced with levels of integration well suited to the implementation of regular storage arrays, but not to the implementation of the multitude of variations encountered during the logical design of a digital system.

The feature of Large Scale Integration which causes the problems is the move towards a large number of circuits on a semiconductor chip. Past levels of technology have enabled the logic designer to use the building block principle, and construct his designs from a number of similar low level packages at card or module level. Variations in design, in this instance, are usually achieved by personalising the interconnections between standard packages, and in this way a range of units can employ a standard set of packages. Furthermore, when design errors and alterations in system specification occur, the system personality is at a level where it can easily be changed.

LSI is typically defined as putting more than 100 logic gates on a chip. This means that the logical function which the chip now carries is so complex that multiple usage of a particular chip design within a system is rare. It implies many different chip types as a result of the high level of personalisation of the chip. Moreover, much of the system personality is now in the form of metallised patterns on the surface of the chip and it cannot therefore be modified in situ. For this reason the introduction of system modifications involves long delays.

The low level package implemented in LSI should retain an acceptably low connector to circuit ratio. Its form should be independent of specific applications as this will enable chip design to continue without reference to the system design cycle. Changes in the user's designs should not require physical alterations of the package at a chip or module level.

Conventional logic methods used with LSI do not achieve these objectives. Alternative ways of exploiting LSI must be found if progress in logic organisations is not to be hindered. The regular store type structure, well suited to large scale integration techniques, provides a pointer to the logic media that will be most readily available in the future. The conventional binary addressed store may be made to perform logic, but is not well suited to this role. The associative array provides a regular structure in which data manipulation is more effectively performed. Previously described methods for processing the data held in an associative array provide a basis from which to work.

Some regularly structured parts of a digital system could possibly be translated directly into LSI (e.g. registers and arithmetic units). Areas least suited to high levels of integration are those having no regular structure, such as those performing machine control. The experimental logic element described has been proposed with this problem in mind.

The techniques described are those of processing data using associative storage. This data is passed against control and logic tables held in the associative arrays. This allows the array to be treated in a way which is analagous to that of the discrete logical element within current system organisations. It results in a wide variety of potential implementations, the efficiencies of which are normally related to the logic designer's skill.

Because the logic of the system is represented in part by the contents of the arrays the efficiency with which this is stored, and used, can have a marked effect on the cost of implementing any given unit. The experimental associative cell employed in the logic array has three states. This type of cell was originally proposed by B. T. McKeever[1] for use with a cryogenic technology. It allows a considerable degree of table minimisation and thus helps to achieve an economical result.

This paper seeks to demonstrate how the logic designer can specify a system design in terms of tables held in interconnected stores.

1. CELL, ARRAY AND BOM

The element of array logic to be described is termed the Functional Memory Basic Operating Module (or FM BOM). The FM BOM contains a writable associative array which replaces the discrete logic element as the medium within which the logic of the system is defined. The array is supported by a number of additional components included within the FM BOM. They are the mask store and mask register, input-output register(s) and function controls. A general layout for the FM BOM is shown in Figure 1.

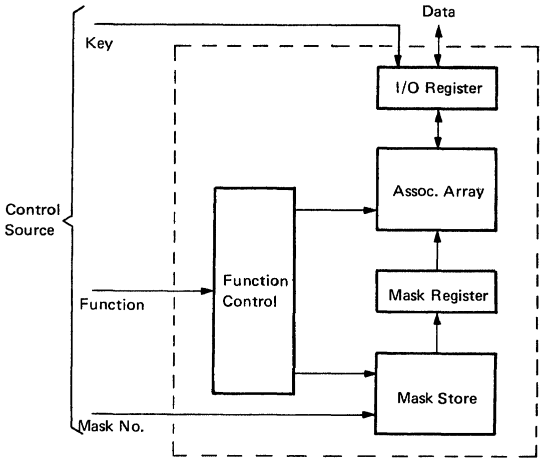


Figure 1. FM BOM

The basic cycle of an FM BOM is divided into two phases: these are the Select phase and the Read/Write phase as shown in Figure 2.

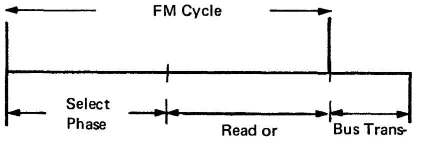


Figure 2. FM Cycle

The type of cell used in the array is shown in Figure 3. The cell has four stable states. Of these, three are normally given meaning in the implementation of the system logic. The cell consists of two similar bistables. Each bistable has a single emitter, which is held at a constant potential, and a dual emitter, by which it is linked to the array. The dual emitters are connected individually to bit and word lines. By suitable manipulation of the potential of these lines, the transistor current can be switched from one line to the other. The path into which the current is switched is dependent on the relative potentials of the bit and word lines. The two halves of the cell are connected to separate bit lines which are termed BIT 0 and BIT 1. The word line is shared. The conventions adopted for the states of the cell shown in Figure 3 are;

0 = T1, T3 on
1 = T2, T4 on
X = T2, T3 on

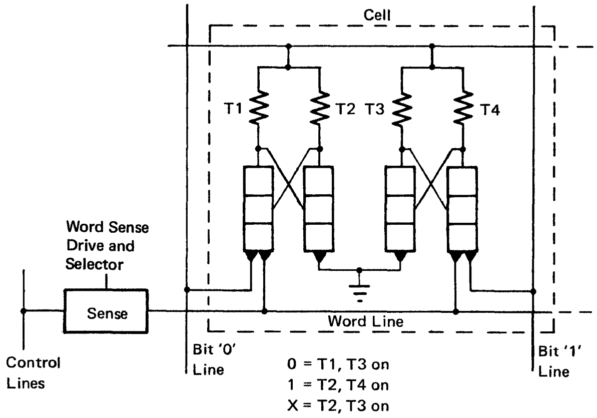


Figure 3. FM Cell

During a Select phase the two bit lines, driven antiphase from a single input-output register position, are used to interrogate the cell. It will be seen that raising the BIT 0 line with a zero stored in the cell diverts current into the word line. Current is also diverted into the word line where the cell contains a '1' and the BIT 1 line is raised. Bit line manipulation has no effect on the word line currrent where the cell state is X.

A match condition is sensed when there is an absence of word line current during the Select phase. The convention is, therefore, that the BIT 1 line is raised when interrogating for a '0' and similarly the BIT 0 line is raised when interrogating for a '1'. A latch per word retains the result of a Search operation upon the word with which it is associated. This latch is termed a 'selector'.

The Read operation is performed from words previously selected in the Select phase. To do this the selected word lines are raised in potential. This diverts current into the bit lines which is then sensed by the bit sense/driver circuits.

The cell Write operation is performed by first raising the appropriate bit line(s) such that they represent the data to be written. When the word line, selected in the Select phase, is subsequently raised in potential the two cell halves assume the state represented by the bit lines.

Figure 4 shows an array in which a cell of the type described is situated at every intersection of bit line pairs and word lines. Each bit of the input-output register has access to its related column in the array via its respective mask gate and bit sense/drive circuit.

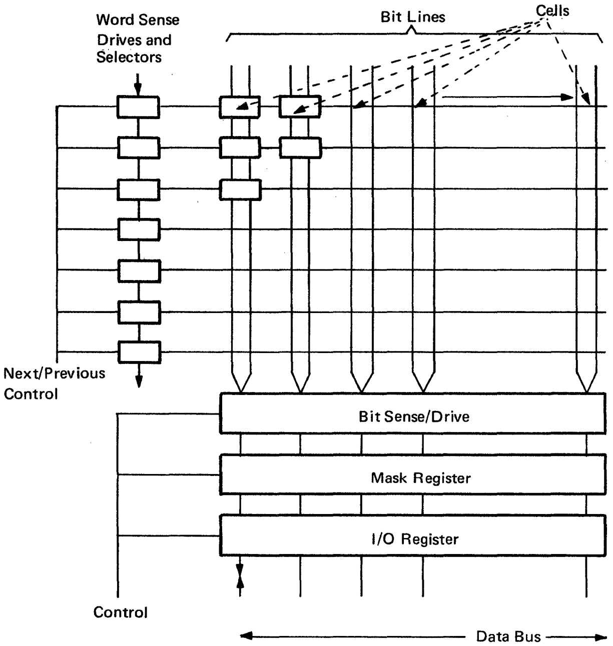


Figure 4. FM Array Layout

During a Search operation the contents of an input-output register are compared with the contents of all the array words within its BOM. The comparison will occur over the positions identified by '1s' in the mask register. In this way the cell columns over which a given Search operation is to be performed may be defined by a particular mask register pattern.

The presence of 'X' states within the array can result in a match condition existing between the contents of the input-output register and multiple words within the array. The selectors for words identified in this way are set to '1' at the end of the Select phase.

The contents of selected words only are taken into consideration when performing array Read. Using this rule the input signal to each bit in the input-output register is derived from the OR of the contents of its related cell column. The transfer from array to input-output register is gated by the mask register. In this way the bit positions, over which a read operation is performed, may be defined by a particular mask register pattern.

The Read operation senses the signals on only the BIT 1 lines and, where they are equal to '1', the corresponding input-output register bits are set. The other positions are reset. This transfer is gated to occur for positions where the mask register bit is a '1'. Because the signals on the BIT 0 lines are ignored, neither the X state nor the zero state in the cell affect the result of the Read into the input-output register.

The Write operation causes data in the input-output register to be set into the words selected during the Select phase. Array columns identified by a mask register bit equal to '1' are the only ones affected by this function.

The mask register pattern is derived from the mask store which contains patterns corresponding to the table formats within the array. This is shown in Figure 1. Separate Select phase and Read/Write phase masks are normally supplied to the mask register by a timing control within the FM BOM. The identity of the mask required is decoded by the FM BOM from a control pattern. Control patterns supplied to the FM BOM from the control source select the required masks together with the Search, Read and Write operations.

Single selector latches per array word are normally employed. In some applications a second group of selectors can provide greater logic flexibility. The selector group used in any FM cycle are defined by the control source. It is normal to link the selectors together within the FM BOM in the form of a shift register. This provides the only fixed logical link between the individual words of the array. A control function, called Next, enables a pattern of set selectors to be shifted one position through the shift register. A function similar to Next, termed Previous, provides a selector shift in the reverse direction. Using the selector shift, adjacent words to the ones currently selected can be read from, or written into, during an FM cycle. Search and either Next or Previous, specified to occur in the same cycle, perform selection of words adjacent to those which match the search pattern.

An FM store of a greater capacity than that defined by a single FM BOM can be formed by one to one linking of the input-output register connections on a number of BOM's. When a number of FM BOM's are bused together in this way the result is a store which has the same logic characteristics as a single BOM.

On completion of a cycle, an FM store will contain identical data in the input-output registers of the constituent BOM's. This is achieved in practice by first setting the individual input-output registers from the local array. Subsequent transfer to and sensing of the bus, results in the OR of the results in individual BOM's appearing in all BOM's. This bus transfer period is shown in Figure 2.

Significant improvement in utilisation of the array can be achieved by the provision of two similar input-output registers within the FM BOM. This BOM organisation is outlined in Figure 5. The contents of either of these input-output registers

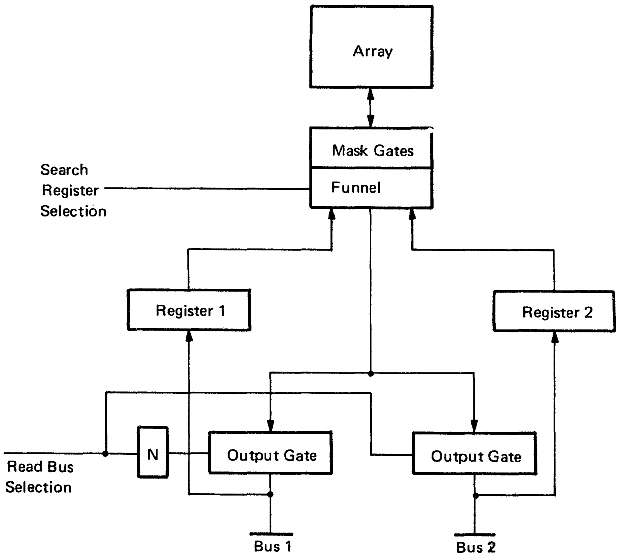


Figure 5. Two I/O Register BOM

may be specified to provide the search pattern. The output of the array may be set into either register and output onto its associated bus. Both registers are set from their respective buses.

The control sequence is as follows:

1. The register specified for the Select phase is used to generate the search pattern.
2. The array is searched and read into the register specified for the Read/Write phase.
3. The read-out from the array is output onto the bus specified for the Read/Write phase and both registers are set from their respective buses.

Note that although both registers are set from their buses only one supplies the search pattern in the next cycle. The register is specified by the control for the next cycle.

2. TABLE DESIGN

Minimising Tables

For FM to be effective as a system technology it should be able to generate any function of its inputs. One way of interpreting the array is as a three stage logic device.

The first stage (the cell) either chooses the true or inverse of the input or ignores it. The second stage (the word) is an AND of the values chosen by the cells in the word. The third stage is an OR of the selected words.

A simple analogy, seen in Figure 6, is able to generate any function of its inputs dependent on the cell states.

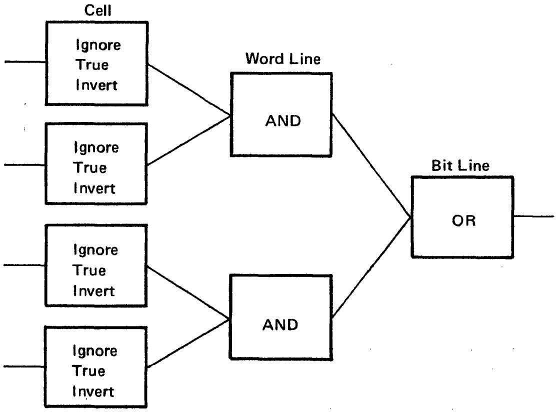


Figure 6. Logic Analogy

When the personality of a system is tabulated in storage, minimising the size of a table is an important design task. Tables are amenable to reduction by the usual logic techniques; by minimising to the disjunctive normal form of the logical expression before conversion into table form. The table in Figure 7 will generate the logical functions of two three-bit operands. The table is searched over the 10 bits of the "table name" A and B fields, and read from the 3 bit result field. The "table name" or "tag" bits are used in combination to select any of the possible 16 logical functions of the two operands. The contents of the I/O register in Figure 7 shows the search fields and the result for an XOR operation.

For clarity the figures illustrating table design use blanks to represent those cells holding the X state.

Note that this table, using FM *3-state* associative cells, requires only 12 words. The same functions, implemented in an associative array composed of *2-state* cells, would require 1024 words.

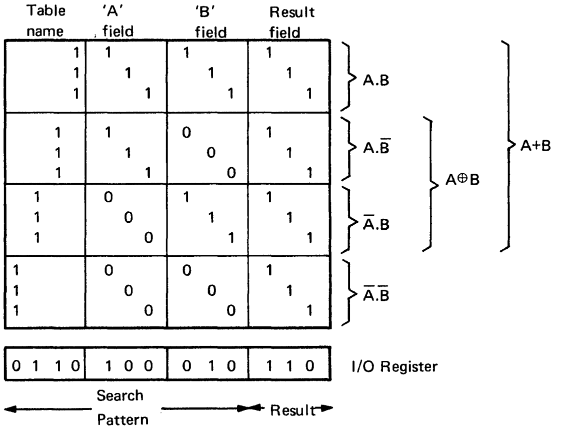


Table name				'A' field			'B' field			Result field			
			1	1			1			1			A.B
			1		1			1			1		
			1			1			1			1	
		1		1			0			1			$A.\overline{B}$
		1			1			0			1		
		1				1			0			1	
	1			0			1			1			$\overline{A}.B$
	1				0			1			1		
	1					0			1			1	
1				0			0			1			$\overline{A}.\overline{B}$
1					0			0			1		
1						0			0			1	
0	1	1	0	1	0	0	0	1	0	1	1	0	I/O Register

$A \oplus B$
A+B
Search Pattern
Result

Figure 7. Complete Logic Table

Reduction techniques have been designed that are particular to FM. These are based on examining the bit patterns in FM, rather than regarding FM as an array into which Boolean expressions may be mapped. Various forms of an increment table will be used to describe these methods.

In Figure 8, four bits form the field to be incremented. The bits are referred to as S8, S4, S2 and S1 in order of decreasing significance. The next five bits form the result field C0, R8, R4, R2 and R1. This table is identical to the truth table for the same function. It performs repetitive increments inefficiently because its output is in a different field from its input. It requires the operand to be re-aligned after every increment.

	Input Field				Output Field					
	S8	S4	S2	S1	C_0	R8	R4	R2	R1	
				0					1	R1=(notS1)
			0	1				1		R2=(notS2).S1
			1	0				1		+S2.(notS1)
		0	1	1			1			R4=(notS4).S2.S1
		1	0				1			+S4.(notS2)
		1		0			1			+S4.(notS1)
	0	1	1	1		1				R8=(notS8).S4.S2.S1
	1	0				1				+S8.(notS4)
	1		0			1				+S8.(notS2)
	1			0		1				+S8.(notS1)
	1	1	1	1	1					C_0=S8.S4.S2.S1.
Search Mask	1	1	1	1	0	0	0	0	0	
Read Mask	0	0	0	0	1	1	1	1	1	

Figure 8. Search Read Increment Table

Using a Search Next operation in the Select phase, followed by a Read operation in the Read/Write phase, the increment table shown in Figure 9 can produce its output in the search operand field. This figure shows how a tag bit is introduced to prevent the matching of the search argument with words intended to output only and also to prevent the subsequent incorrect reading of input words. Odd-numbered words (tagged 0) are searched for, and even numbered words (tagged 1) are read out. Note that use of the Search Next Read operation doubles the number of words in the increment table from 11 to 22 words, but almost halves the table width.

			Input and Output Operand Field				Word No.
	Tag	Co	S8 R8	S4 R4	S2 R2	S1 R1	
	0					0	1
	1					1	2
	0				0	1	3
	1				1		4
	0				1	0	5
	1				1		6
	0			0	1	1	7
	1			1			8
	0			1	0		9
	1			1			10
	0			1		0	11
	1			1			12
	0		0	1	1	1	13
	1		1				14
	0		1	0			15
	1		1				16
	0		1		0		17
	1		1				18
	0		1			0	19
	1		1				20
	0		1	1	1	1	21
	1	1					22
Search Mask	1	0	1	1	1	1	
Read Mask	0	1	1	1	1	1	

Figure 9. Search Next Read Increment Table

The number of words can be reduced, without increasing table width, by allowing words to be both searched for and read-out. For example, in Figure 9, word 3 is a searched-for word; it contains 1's in the same position as word 2, which is a read-out word. Word 2 may therefore be deleted, and word 3 used as the read-out word for word 1. Words 4, 6, 8, 10, 14, 16, 18 may be similarly removed. With these deletions, the 22 word table in Figure 9 shortens to the 14 word table shown in Figure 10.

			Input and Output Operand Field				
	Tag	C_0	S8 R8	S4 R4	S2 R2	S1 R1	Word No.
	0					0	1
	0				0	1	3
	0				1	0	5
	0			0	1	1	7
	0			1	0		9
	0			1		0	11
	1 (Not Selected)			1			12
	0		0	1	1	1	13
	0		1	0			15
	0		1		0		17
	0		1			0	19
	1 (Not Selected)		1				20
	0		1	1	1	1	21
	0	1					22
Search Mask	1	0	1	1	1	1	
Read Mask	0	1	1	1	1	1	

Figure 10. Minimised Increment Table

This reduction method may be applied to most Search Next Read tables, if not directly, then after word re-ordering.

Words Shared between Functions

Certain assignments of tag values allow words to be shared between functions. A simple example of this is the logic function table of Figure 7, where the 16 logic functions share the words of the table.

Any single table may be used for more than one function by searching with different masks, and by operating the FM with Search Read or Search Next Read. The advantage of sharing words, or tables, is that the number of words used is reduced.

As an example of this the table in Figure 11 may have been generated for the function 'move A to B, maintain C', where in this case A, B and C are single bit fields. The table is searched over the field 'tag AC' and is read over the field 'BC'. If the tags in this table are changed to XI and IX, as in Figure 12a, then it is possible to perform not only 'move A to B and maintain C', but also other functions, some of which are listed in Figure 12b. The function performed in any one cycle then depends upon the choice of search mask, read mask, type of search operation and value of key (input pattern used to search the tag field).

Tag	A	B	C
11	1	1	
11			1

Figure 11. Logic Table

Tag	A	B	C
X1	1	1	
1X			1

Figure 12(a). Improved Logic Table

Function	Search Mask	Read Mask	Key	Operation
A to B	Tag A	B	01	Search
B to A	Tag B	A	01	Search
A to C	Tag A	C	01	Search next
C to A	Tag C	A	10	Search previous
B to C	Tag B	C	01	Search next
C to B	Tag C	B	10	Search previous
B to A, C to C	Tag BC	AC	11	Search
A to B, C to C	Tag AC	BC	11	Search
A to A, C to C	Tag AC	AC	11	Search
B to B, C to C	Tag BC	BC	11	Search
A.B to A, C to C	Tag ABC	AC	11	Search
A.B to B, C to C	Tag ABC	BC	11	Search
A.B to A, A.B. to B, C to C	Tag ABC	ABC	11	Search
A.B. to C	Tag AB	C	01	Search next
C to A, C to B	Tag C	AB	10	Search previous

Figure 12(b). Logic Table Function List

Some of the above functions may Search and Read over wider fields than those shown: for example the 'A to B' function might use search mask 'tag AC', read mask 'BC', without altering the function.

In the above example a two word table was shown to provide fifteen different functions. Most tables provide a multiplicity of functions and this factor can be exploited in optimising a design.

Table design starts with a requirement for a set of functions and ends with bit patterns in several ordered consecutive words. Generation of bit patterns involves using the minimum number of adjacent words to give the required functions. The bit patterns could be generated by computer program, using as input the required combination of functions together with the FM tools available, but for some applications an ad hoc method may be found to provide better optimisation. The bit pattern structure differs in concept from the truth table interpretation that could be placed on the increment table described in Figure 8.

3. SYSTEM DESIGN

Control of FM Stores

An FM Store in a data flow is controlled by a field emitted from a store (which may be an FM Store). This field has three parts: Function, Mask and Key. These inputs are seen in Figure 1.

The most primitive FM operations are Search or Next, which are operations on the selectors, and Read or Write which are operations on the selected words.

In practice Select and Read/Write phases tend to alternate because successive primitives such as Search Search or Write Write are redundant.

It is therefore convenient to define a composite function rather than use the primitives. The composite function has two phases as shown in Figure 2. The first is the Select phase, in which Search, Next, Previous and other operations on the selectors may be specified. The second is the Read/Write phase in which the selected words are either read or written.

The advantages of the composite function are:

1. The number of control words used is halved because in the environment of table look-up data flows, a Select phase now precedes each Read/Write phase.
2. An FM store is able to control another FM store, or itself, because it allows an output as a result of each control input.
3. The transfer of data between two FM stores is simplified if they are synchronised by phase. Any speed advantage given by a sequence of primitives over a sequence of composite functions is lost in synchronisation when transferring the result to a second store.

The part of the control field specifying the composite function is typically 4 bits wide.

1. Search
2. Next
3. Previous
4. Read/Write

The first bit specifies Search and if no Search is specified then the selectors retain their setting from the preceding cycle. If a Search is specified Bits 2 and 3 define the selector shift operation which is to follow.

Possible selector shift operations are:

no shift
shift Previous
shift Next

Bit 4 specifies Read or Write.

The second part of the control field addresses the mask store. The widths of the FM store over which searching, reading or writing occur is defined by a mask. The mask address specifies two masks, one to be used during the Select phase and the other to be used during the Read/Write phase.

A field of the FM store is reserved for the tag as shown in Figure 13. Each word of the table contains the same tag value in this field. Every table must have a tag and different tables have different tag values. To select a table the store is searched using a key pattern which will match the tag for the required table. The key is supplied to the FM store from the third part of the control field.

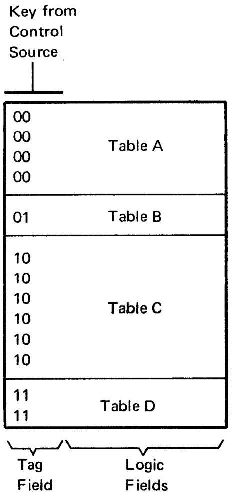


Figure 13. Table Addressing

A field of the FM bus can be returned to the control store as a micro-instruction address modifier to allow for conditional execution of microprogram (condition field). Figure 14 shows the organisation of such a control system.

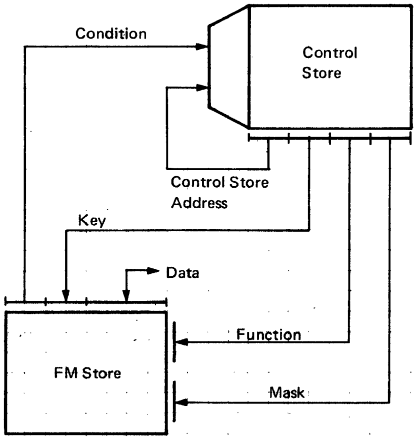


Figure 14. Control of an FM Store

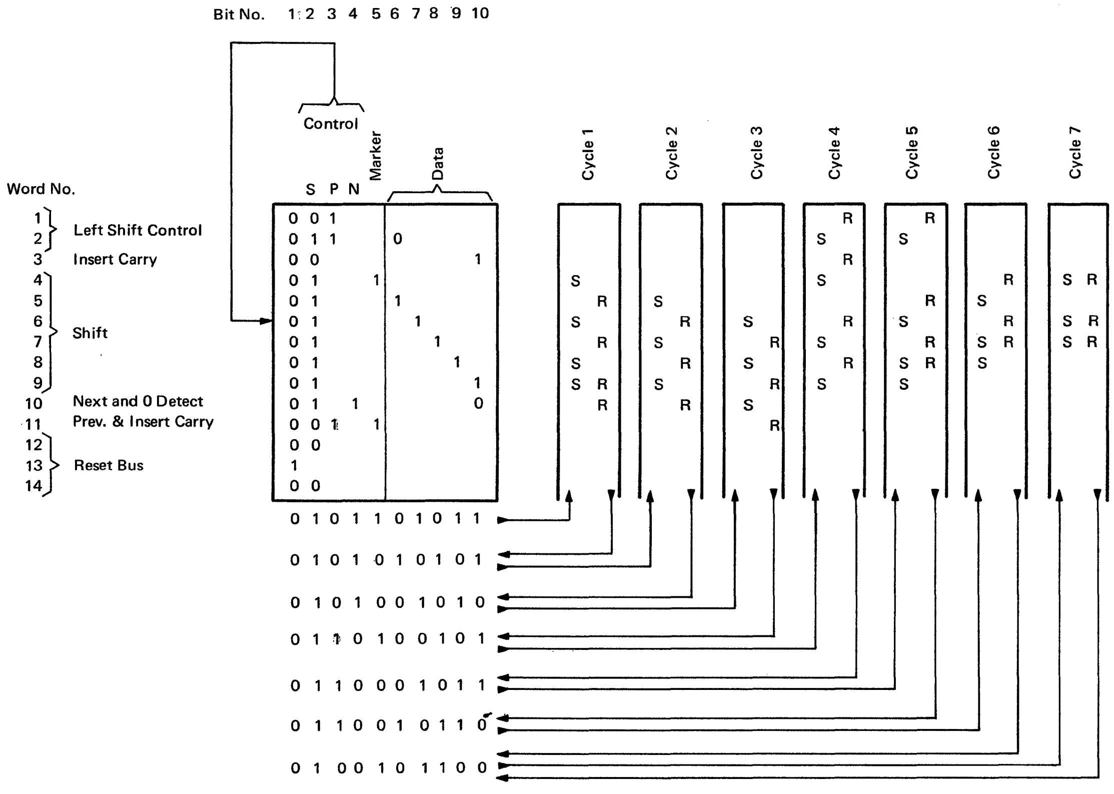


Figure 15. Auto Sequencing

Auto-Sequencing

Certain system designs require that a part of the system operates independently for a period, e.g. an I/O adapter. The principal function performed is generally simple (transfer of a data block), with intermittent need for more complex operation (test the status of the transfer). The cost of an FM based system can be reduced if those partly independent areas can generate their own control patterns for the simple sequences, and share a control for the complex sequences. The auto-generated pattern may be all, or part of, the control input to the FM store. A store, containing a table to increment by '1' the contents of a five bit field, is shown in Fig.15.

This store is able to specify its own Search, Previous and Next controls by reading onto bus bits 2,3 and 4 respectively. The table increments the five bit field defined by bits 6-10 by right shifting with the Search Next Read function. This is repeated until the least significant bit is zero, whereupon a '1' is inserted into that bit position. Then the shift reverses (using Search Previous Read), zeros being shifted in until the field is realigned. The table then maintains the result (Search Read).

Bus bit 5 is used as a marker position. It is initially set to '1' to provide a frame bit to delimit the data. It is also used to hold the carry when the shift reverses.

Words 1 and 2 control the left shift.

Word 3 places the carry in position.

Words 4 to 9 form the shift.

Work 10 generates the Next control and detects a zero in the least significant position.

Word 11 generates the Previous control and places the carry in the marker position.

Words 12, 13, 14 are used to reset the bus. A Search and Read with a '1' in bit position 1 resets the control and data fields. This operation, followed by a Search Next Read with a '1' in bit position 1, will reset the bus.

In cycle 1 of the example the external control loads the data and initiates the increment by inserting a 1 into bits 2 and 4. Thereafter the FM is auto-sequenced.

In cycles 2 and 3 the data is shifted right to shift out the least significant 0.

In cycle 4 the data is shifted left and the carry introduced.

In cycles 5 and 6 the data is realigned by shifting left.

From and including cycle 7, the data is maintained on the bus until the store is reset by external control.

This example demonstrates not only auto-sequencing but also how control has been assimilated into a table in an array.

A store can be considered to be auto-sequencing if it modifies its own control, mask or key, or any combination of these. Auto-sequencing is a technique that can be applied to more complex functions than those shown. The major limitation of auto-sequencing is the need for regeneration of control following a Write operation.

Pipeline

The execution of functions can be spread over several cycles using tables which generate components of the function. In the case of addition, the use of the factors XOR, CARRY

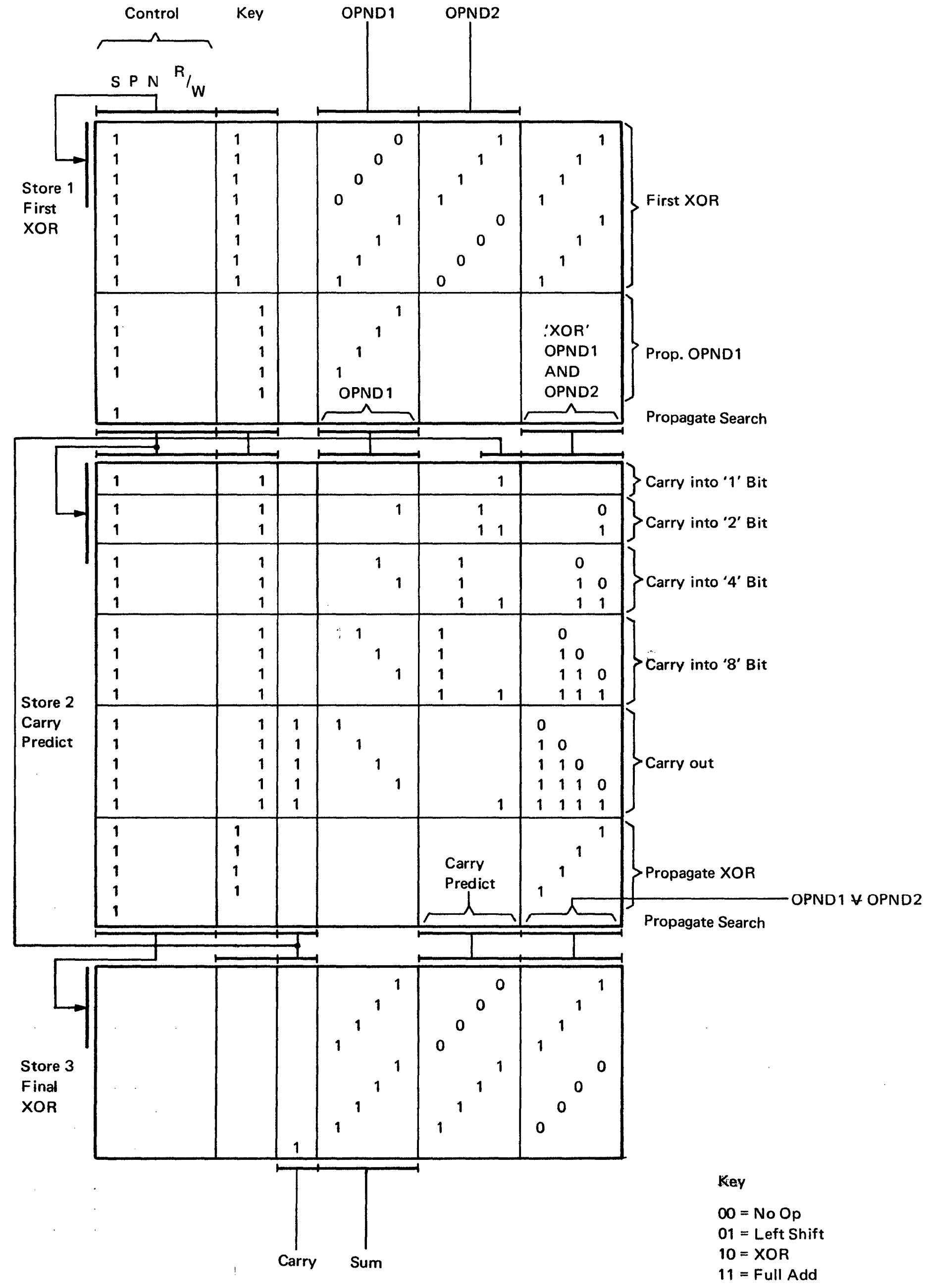

Control
Key
OPND1
OPND2
S P N R/W
Store 1
First
XOR
First XOR
'XOR'
OPND1
AND
OPND2
Prop. OPND1
OPND1
Propagate Search
Carry into '1' Bit
Carry into '2' Bit
Carry into '4' Bit
Carry into '8' Bit
Store 2
Carry
Predict
Carry out
Carry
Predict
Propagate XOR
OPND1 ⊻ OPND2
Propagate Search
Store 3
Final
XOR
Carry
Sum
Key
00 = No Op
01 = Left Shift
10 = XOR
11 = Full Add

Figure 16. Pipeline

PREDICT, XOR reduces the number of words used from

$$2^{n+3} + 2^{n+1} - 4n - 9 \text{ to } \frac{n^2}{2} + \frac{15n}{2} + 5$$

In the case of byte addition the number of words used is reduced from 2519 to 97.

These partial function tables can either be in a single store, or in separate stores which form a pipeline. Each of these methods uses the same number of words. The pipeline scheme can give three times the throughput when fully utilised.

A three stage pipeline which performs a digit add is shown in Figure 16. Each stage is an FM store consisting of FM BOM's each having two I/O registers.

Store 1 forms the XOR of operands 1 and 2, which is passed, with operand 1, to Store 2.

Operand 1 and the XOR of Operand 1 and 2 are used by Store 2 to generate a carry predict pattern which is passed with the XOR to Store 3. An interconnection between the lower and upper I/O registers provides a path for carry propagation between digits.

Store 3 forms the sum by XORing the XOR and Carry Predict patterns. Control and key for Stores 2 and 3 are generated by the operation in Stores 1 and 2, respectively.

The tables in Stores 1 and 2 have been tagged so that the pipeline will perform four functions. These are chosen by the keys presented to store 1:

00 NO OP
01 LEFT SHIFT OPERAND 1
10 XOR OPERAND 1 AND 2
11 ADD OPERAND 1 AND 2

This example demonstrates how stores can be interconnected to form a data flow.

CPU

Figure 17 shows how three FM stores might be interconnected to form the data flow of a CPU. Also in the diagram is a conventionally addressed Main Store and a microprogram control store.

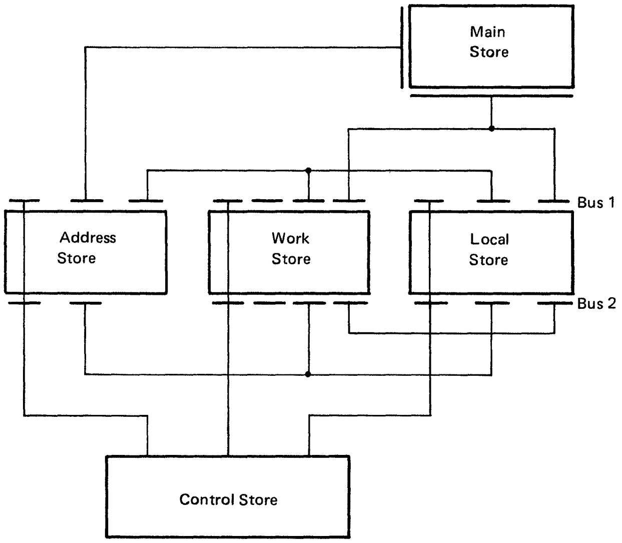


Figure 17. Data Flow

The operations performed by each FM store can be only broadly defined because, as might be expected, each store contains a mixture of tables used for logic operations, for storage and for control and decode purposes. In other words the basic functions of a data flow are much more diffused than in a conventional data flow. This can be illustrated by the way in which data patterns, exceptional conditions and decoding are recognised. In a conventional logic data flow these are very often recognised by matching data or conditions against a set of stored constants, using the arithmetic unit of the data flow to select a path through a microprogram decision tree.
Using FM it is generally cheaper and faster to form condition sensitive tables which directly detect the pattern. These tables are placed so that they operate on the data at its source in the data flow. Data need not be routed around the data flow to a single localised condition or decode unit.

The functions of the three FM stores may be broadly categorised as follows:

The Address Store provides the Main Store addressing path; it provides storage for addresses and an increment table which can be used in parallel with other arithmetic operations in the data flow.

The Work Store holds tables for basic data manipulation, for instance addition, subtraction and logical operations.

The Local Store provides working registers for temporary data storage.

It must be emphasised that these distinctions are a matter of degree. In fact all three stores perform some function of condition detection, instruction decoding, data storage and data manipulation.

4. CHECKING AND AVAILABILITY

Checking

The physical and logical characteristics of the functional memory logic element determine the checking methods used in units of which it is a part. Its structure, in which logic tables can operate correctly without being tied to specified locations in a store, suggests significant potential for greater systems availability.

It will be noted that a major feature of the array is the possibility that multiple word selection will result from a Search operation.

Normal checking methods, using parity applied to a word or field of the array, are suitable for instances of single word selection. They cannot be used where the OR of multiple words generate the result. This is due to the lack of a simple relationship between the parity of individual words and the data read from the array into the input-output register.

It is seen, therefore, that because of multiple word selection, simple parity checking methods are unsatisfactory where a three state cell is employed. An alternative way of using parity is to combine parity prediction tables and function tables within the array. This approach uses a considerable table volume within the FM stores for prediction, and requires input-output register positions to be allocated for check bits. It results in more complex designs and implies restrictions in table structure.

An alternative to the above methods is outlined in Figure 18. The FM BOM is designed with two identical arrays. The checking

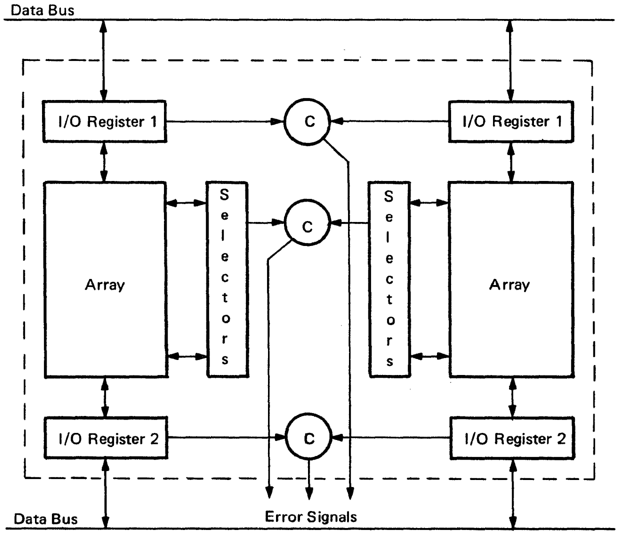


Figure 18. Checked FM BOM

function is performed by comparators (C) monitoring the selectors and input-output registers. In use the two arrays are loaded with identical tables and the comparators check each cycle for agreement. An error condition results in a compare check.

Checking by duplication in the FM BOM enables error detection to be performed independently of the function installed in the tables.

Availability

The duplication, which is included in the FM BOM for checking, can also be used to improve system availability. The checking method described can detect an error in any bit position of the data flow, but will not immediately indicate its cause. When the cause of the error can be determined there is a unique potential for self repair.

The error may be caused as a result of a transient condition within the FM data flow, which can result in spurious changes in the stored tables. Where this is the case, restoration of the tables within the arrays enables correct operation to be resumed. Diagnostic techniques can isolate the fault to an FM BOM and indicate which of the duplexed arrays contain the faulty data. The repair method is to transfer good table data from the duplicate array into the array in which the error is detected.

When the control function is a Search the correct operation of the table in the FM Store is independent of the location and ordering of the constituent words. Similarly a table employing Search Next may appear in any group of adjacent words. These properties can be used to allow the repair of permanent array failures within a store. This is achieved by transferring either words or tables from the faulty area to spare locations reserved for the purpose.

A practical implementation of this technique is shown in Figure 19. It consists of allocating a spare FM BOM to an FM store in which repair facilities are required. Persistence of an error in a specific location is normally the criterion for employing the spare FM BOM. Table data from the good half of the failing BOM is transferred to both sides of the spare BOM and it becomes an operational part of the store. The faulty component is removed logically. The Figure shows BOM's A, B and C, which form the normally used part of the store prior to error. Activation of the error line results in alteration of the pattern in the configuration register, which then indicates the new store organisation.

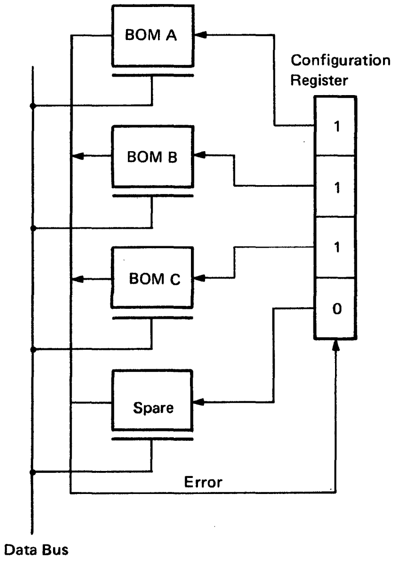


Figure 19. Repair Using a Spare BOM

CONCLUSIONS

A requirement of a system building block is generality of application. This paper is intended to show how the functional memory (FM) concept satisfies this criterion. Any evaluation of a new technology must not be based on a specific operation such as addition, but rather on the way in which it performs the total system function.

The functional memory approach introduces new standards of acceptability in which the increased number of circuits, required to implement certain system functions, must be balanced against significant benefits. These are the provision of a single module of logic which can be personalised by changeable tables, and combines a high level of checking with intrinsic repairability.

The definition of the system logic in a group of writable stores is an important departure from current practice. It provides for systems in which the logic can be implemented using many fewer part numbers than are possible today.

The function of an FM system can be modified, without physical change, by alteration of the table structure. This is useful in the design phase, where errors, and changes in system requirements, can be easily contained. It simplifies the engineering change problems in the field and gives greater scope for designs personalised to a users needs.

It is likely that with reductions in the costs of logic circuits there will be more emphasis on improving system availability. The results will be greater user satisfaction and lower maintenance costs. The capabilities of FM for performing self repair and restart are significant in this context.

Systems built in current technologies normally use diagnostic methods which are specific to each unit type. The reason is that fault location, often imprecise, requires a detailed knowledge of the logical structure being diagnosed and, further, uses this structure to assist in the diagnosis. Where logic is held in a writable array this situation changes. Testing the integrity of a logic element in a system need not now be tied to its functional use. The diagnostic procedures can be similar to those used for other memory structures, and can operate using different stored data from that required for functional operation. A benefit here is that there are now separate methods of diagnosing component failures and design faults.

The checking, by duplication built into the FM BOM, simplifies the problem of identifying a faulty module. This should reduce both equipment repair time and the degree of training required by field personnel. It should also reduce the cost of module testing during manufacture.

It is felt that current systems have been shaped by the characteristics of the available logic. This study has shown that the use of a technology with very different attributes can lead to new concepts of design. Such concepts can lead to new ways of appraising designs and the result could be a marked change in the systems of the future.

REFERENCES

1. B. T. McKeever, Proceedings – Fall Joint Computer Conference 1965.

A MODULAR PLATED-WIRE ASSOCIATIVE PROCESSOR

L.C. FULMER AND W.C. MEILANDER
Goodyear Aerospace Corporation

ABSTRACT

An associative processor is described that combines the content addressability of an associative memory with arithmetic and input-output capability at each word of memory.

The associative processor is classified as an array processor with processing rates from 15 million to 1 billion instructions per second and input-output (I/O) rates from 1 to 160 billion bits per second. Since it is constructed of inexpensive memory modules and has processing rates in excess of the supercomputers, it is projected to be 30 times as cost effective as machines of the ILLIAC IV class.

Due to its content addressability and parallel instruction execution capability, programming the associative processor is postulated to be even simpler than programming a conventional processor.

Finally, a 128-word associative processor demonstration model is described. The test results show that even this small associative processor is capable of operating at 4 million instructions per second.

INTRODUCTION

The internal memory of the conventional digital computer is the dominant element that limits its maximum instruction execution rate and data transfer rate. This memory must be accessed sequentially by location, and it performs no processing on the data that it stores or transmits.

Parallel accessed, content addressable memories (CAM), which process data in memory, promised to overcome some of the limitations of the conventional memory and were seriously investigated in the early 1960's. Several concepts were reduced to hardware, and one system* that is currently in operation for experimental purposes is shown in Figure 1. These early models proved the feasibility of associative memories and demonstrated that logical searches such as equality within limits and extremes could be performed on all words in the memory within one or two cycles of the memory. However, they did not in themselves overcome all of the limitations of the conventional processor.

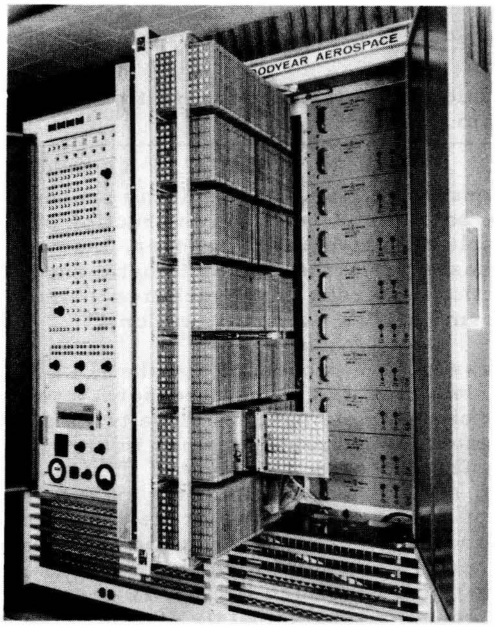


Figure 1 - 2048-Word Associative Memory

The associative processor described in this paper is an evolutionary product of associative memory technology. The associative processor differs from and is more powerful than an associative memory in that it has arithmetic and input-output capability as well as storage and logic capability at each word of the memory. It is, therefore, an array processor capable of parallel arithmetic and parallel I/O operations, in addition to the parallel search operations of the associative memory.

CLASSIFICATIONS

To provide more efficient processing, various machine organizations were proposed and built during the 1960's. These included multiprocessors, array processors, and parallel processors. One convenient way to classify these processors in terms of instruction streams and data streams was proposed by Flynn.[1]

Table I lists the types of processors and some examples that fit Flynn's classifications. Reference 1 does not consider the associative memory or the associative processor, but they both fit the single instruction-multiple data classification and are included as examples of array processors.

Both the associative memory and the associative

---

*2048-word associative memory delivered to Rome Air Development Center, Contract AF30(602)-3549 (1968).

TABLE I - CLASSIFICATION OF PROCESSORS

Classification	Type	Examples
Single instruction stream single data stream (SISD)	Sequential processor	Contemporary digital computers including CDC-STAR, 7600, 6600 IBM-Stretch, 360/91, 360/195
Single instruction stream multiple data stream (SIMD)	Array processor	Solomon machine ILLIAC IV Associative memory Associative processor
Multiple instruction stream single data stream (MISD)	Streaming processor	. . .
Multiple instruction stream multiple data stream (MIMD)	Multiprocessor	Holland machine Univac - LARC

processor are classified as array processors, since both execute a single instruction stream on multiple data streams. However, the associative processor performs both arithmetic and search instructions whereas the associative memory can perform only the search operations.

The associative processor and the ILLIAC IV are both classified as array processors, even though the processing elements of these two machines represent the current extremes in array processing. The processing element of ILLIAC IV is, in itself, a sophisticated computer except that it is void of local control. The processing element for the associative processor, on the other hand, is a rudimentary computer also void of local control. The hardware for each ILLIAC IV processing element contains approximately 10,000 gates and a 2048-word × 64-bit solid-state memory. The hardware for each associative processor processing element consists of 32 gates and a one-word × 256-bit memory. Furthermore, while ILLIAC IV is being designed for a maximum of 256 processing elements (due to obvious economic constraints), the associative processor is easily constructed in multiples of 256 processing elements since it basically is a memory processor.

BASIC PROCESSING ELEMENT

The basic processing element of the associative processor is shown in Figure 2. It consists of one word of storage "n" bits in length and a single processing unit called a response store.

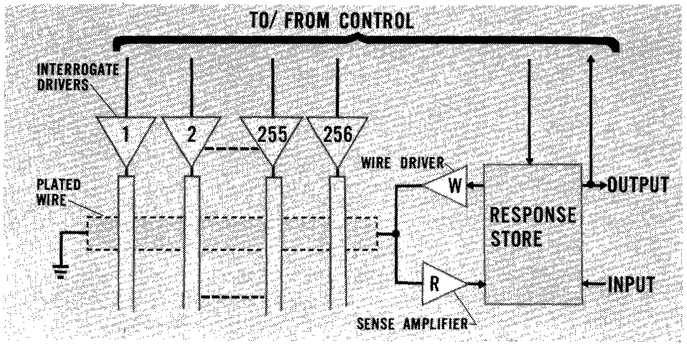


Figure 2 - Basic Processing Element

For the specific organization discussed in this paper, the one-word memory is a single-plated wire that is 256 bits in length. This one-word memory has the capability of being written either in a bit parallel (32 bits) mode or in a random access bit serial mode. It also has the capability of being read (nondestructively) in a random access bit serial mode.

The 32-bit parallel write allows the control unit to write conveniently into the memory. The random access bit serial write-and-read mode performs two important functions. First, it allows the control unit, operating with the response store, to execute instructions at each processing element. Secondly, it allows the outside world to communicate with any bit of the memory word through the response store.

The major function of the response store is to provide logic and buffer storage for each word of the associative processor. In conjunction with the control unit, it functions as a serial arithmetic unit, a one-bit store for responders to a search, a buffer for parallel input-output, and controls the writing into memory. Since this logic exists for each word of the memory, optimum design is important. Speed versus logic tradeoffs are a function of both the memory element of the associative memory and the circuit elements of the response store.

Using plated wire, hybrid drive, and sense circuitry for the memory element and a 32-gate MSI chip for the response store results in a processing element capable of read and write speeds of 100 nanoseconds and 300 nanoseconds, respectively. This results in an exact match search time of 100 nanoseconds per bit and an add time of approximately 500 nanoseconds per bit.

ASSOCIATIVE PROCESSOR MODULE

Figure 3 shows a 256-word associative processor module. For a plated-wire associative processor, this is merely an array of 256 plated wires and 256 sets of response stores. The interrogate straps encompass all of the plated wires and provide the means for a parallel interrogation of any one bit of all 256 words in the memory.

TABLE II - ASSOCIATIVE PROCESSOR PROCESSING RATES

Processor size		Processor rates		
Number of modules	Number of processing elements	Instruction execution rate (inst/sec)	Output transfer rate (bits/sec)	Input transfer rate (bits/sec)
1	256	15.8×10^6	2.56×10^9	0.85×10^9
2	512	31.6×10^6	5.12×10^9	1.71×10^9
4	1,024	63.2×10^6	10.24×10^9	3.42×10^9
8	2,048	126.4×10^6	20.48×10^9	6.84×10^9
16	4,096	252.8×10^6	40.96×10^9	13.68×10^9
32	8,192	505.6×10^6	81.92×10^9	27.31×10^9
64	16,384	1011.2×10^6	163.84×10^9	54.61×10^9

sum back in the respective processing element. The I/O rates are based on delivering 256 bits in 100 nanoseconds and on accepting 256 bits in 300 nanoseconds for each module.

Table II shows that the modular associative processor has raw processing rates from 15 million to 1 billion instructions per second depending on the number of modules in the processor. These figures give some idea of the tremendous inherent capability of the associative processor.

ASSOCIATIVE PROCESSOR VERSUS THE SUPERCOMPUTERS

The associative processor offers a practical, cost-effective solution to the demand for machines capable of greater data processing speeds.

To meet the current demand for higher processing rates, superscale machines such as Control Data's STAR, 7600, and 6600; IBM's 360/91 and 360/195; and the University of Illinois' ILLIAC IV have been or are being built.* These machines cover a range of processing rates from 2 to 200 million instructions per second (MIPS) at a cost to the user of between 3 and 10 million dollars. The most powerful of these, ILLIAC IV at 200 MIPS, has a projected cost estimated at 6 million dollars per quadrant.** For comparison purposes, then, the ILLIAC IV cost per MIP is \$30,000. The associative processor, on the other hand, has a projected cost of \$15,000 for a module capable of 15 MIPS, or \$1000 per MIP. Since the associative processor can cover the range from 15 to 1000 MIPS by modular expansion, it could perform at the same rate as ILLIAC IV but at 1/30th of the cost.

SOFTWARE CONSIDERATIONS

Current industry estimates indicate that software costs for conventional computer applications range up to or equal the hardware costs. For the supercomputers and/or special real-time systems, this ratio of software to hardware costs might escalate even higher.

Programming complexity and costs are roughly proportional to the number of instructions required to execute a given task. In all detailed application studies conducted by the authors and their colleagues, the associative processor required significantly fewer instructions to execute a given task than did the conventional computer system. Although many of these studies are classified, Reference 2 gives some unclassified results. This reference shows that in one part of the problem the associative processor system required only eight percent of the instructions required by the conventional system. For the complete solution, the associative processor required only two-thirds the instructions required by the conventional system.

Given that a high degree of parallelism exists in the problem, the associative processor will require less instructions to perform a given task than any other processor. It will, therefore, cost less to program the associative processor than the conventional processor.

The associative processor has two extremely powerful features that simplify its programming: (1) parallel content addressing and (2) simultaneous instruction execution on the entire data base.

Parallel content addressing replaces serial location addressing and all the bookkeeping problems that go with it. Data can be stored and retrieved in the associative processor without regard for its physical location in memory. Ordered lists, file maintenance, hash addressing, binary searches, tree structures, and other programming aids used to speed up the operation of the conventional machine are not required in the associative processor. Parallel content addressing reduces the problem of data access to a straightforward parallel search.

The ability of the associative processor to execute

---

*Datamation, p 127, November 1969.

**Knapp, M.: Comments at the Symposia on Parallel Processor Systems, Technologies, and Applications, Monterey, Calif., June 25, 1969.

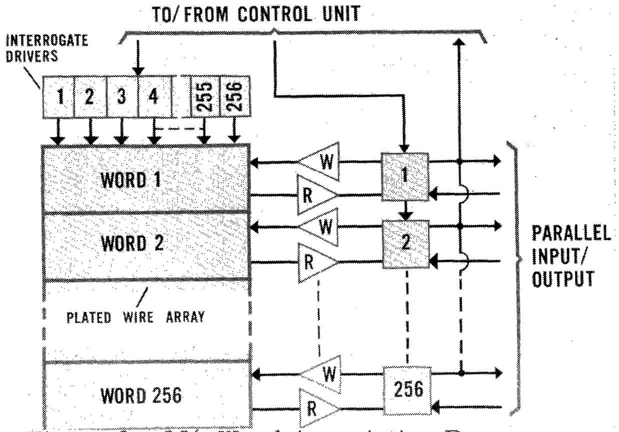


Figure 3 - 256-Word Associative Processor Module

This 256-word × 256-bit module is the basic modular building block of the associative processor. It is designed and constructed so that a higher capacity processor can be realized merely by adding more of these modules.

The 256-word associative processor module has several features that are not readily apparent when only the basic processing element is considered. These include problem partitioning, communication with neighboring elements, and communication with the control unit.

Problem partitioning is provided through the content addressability of the module. Besides allowing data to be stored without regard to ordering, this feature allows different classes of data to be stored in the same module as long as it is identified by class using tag bits. In operation, the control unit would execute a parallel equality search on the tag bits and then execute a single instruction stream in parallel on only those words that responded to the equality search.

Communication between neighboring elements is provided by a shift up or down capability in the response store. This feature is useful in extending the word length of a processing element (for example, variable word length). It is also the vehicle that allows the associative processor to perform arithmetic operations between processing elements rather than confining the operands to be within the processing element.

Communication with the control unit is provided by "OR-ing" the output of all response stores and supplying a single output line called the sum-OR. This line is used by the control unit to (1) determine if there are any responders, (2) read data from selected responders, (3) determine count of responders, and (4) determine address of responders.

There are two sets of input control lines from the control unit. One set is distributed to the 256 memory interrogate straps, and the other set is distributed to common busses for response store control. Also, since each word of the memory is provided with an input and an output line, the memory module can operate at very high data rates in a parallel I/O mode. For example, a 256-word module can output 256 bits every 100 nanoseconds and can accept 256 bits every 300 nanoseconds.

THE COMPLETE ASSOCIATIVE PROCESSOR

The complete associative processor consisting of "M" modules under global control of a single control unit is shown in Figure 4. The control unit is of conventional design and can be functionally divided into the sequential control portion and the associative control portion.

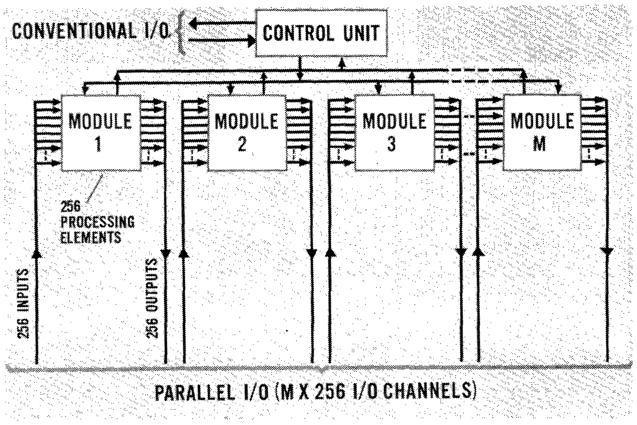


Figure 4 - Associative Processor (MX256 Words)

The sequential control portion basically is a conventional processor complete with sequential input-output, arithmetic unit, and a location addressed memory. The primary function of the sequential control portion is to process and store the instructions to be executed by the associative processor modules.

The associative control portion is a set of registers, counters, and control line drivers and receivers that interface with and control all associative processor modules. Since at any time only a single instruction is being executed on all modules, the counters and registers are common to all modules. Only the control line drivers and receivers increase as the number of modules increase.

The practical upper limit on the number of modules (M) would be determined by the physical properties of the hardware in conjunction with the speed requirement for the processor. Since the plated-wire modules are only about 0.5 cubic feet, M easily could be 64 or 128. But even an associative processor of 16 modules (4096 words) has a raw processing rate up to 250 million instructions per second. Since a parallel input and output is available for each word of the memory, this processor also has a tremendous capacity to accept or transmit data.

Table II summarizes the raw processing rates and I/O transfer rates for various associative processor sizes. The processing rate is based on the time required for all available processing elements to add two 32-bit numbers and to store the 32-bit

TABLE III - TYPICAL ASSOCIATIVE PROCESSOR INSTRUCTIONS

Type/definition	Execution[*] time (microseconds)
Search	
Exact match	0.1(n + 4)[+]
Less than	0.1(n + 4)
Greater than	0.1(n + 4)
Maximum value	0.2(n + 2)
Minimum value	0.2(n + 2)
Next lower	0.3(n + 3)
Next higher	0.3(n + 3)
Arithmetic	
Add common	0.8(n + 1)
Add fields	1.1(n + 1) + 0.1
Multiply common	0.8b(a + 3.2)
Multiply fields	1.1b(a + 2.7)

[*] Execution time is independent of number of words on which algorithms are performed.

[+] n, a, b = number of bits in pertinent data field.

The radar scan time limits the time available for processing so that a real-time situation exists. Each set of target coordinates seen by the radar represents an independent data set that is to be processed, and collectively the target parameters represent a multiple data stream.

A typical radar processing, track-while-scan, and display system would have a functional flow diagram as shown in Figure 6. Each box is discussed very briefly to keep the paper within reasonable length but so the reader gets a look at associative processor application.

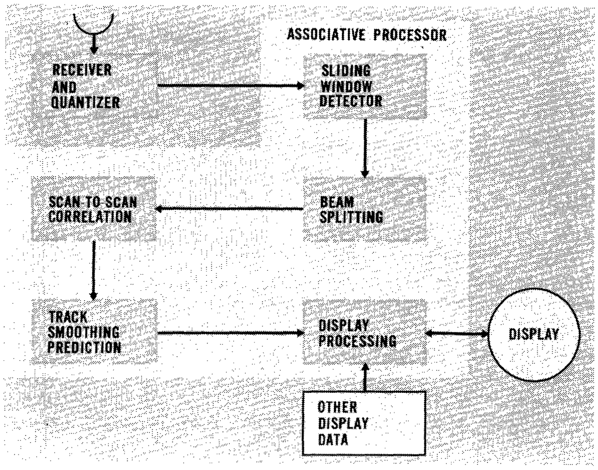


Figure 6 - System Surveillance Functional Flow Diagram

The sliding window detector performs the first decision process. The process begins by summing all positive threshold crossings at a specific range increment over a number of pulses equal to the antenna beamwidth. The same sum is developed for all range increments within, say, plus or minus 5 of the range under consideration. The sum of all 11 range intervals is developed. If the sum of the specific range interval is greater than a value derived from the sum of all 11 range intervals over the same beamwidth, a target is declared.

This operation is continued in range by "sliding" the window up one range increment and repeating until maximum range is reached. The operation is continued in azimuth by bringing in the next pulse from the quantizer; the process is repeated as scanning proceeds.

Radar systems familiar to the authors that use the sliding window detectors are special-purpose dedicated devices occupied 100 percent of the time. In one example studied, a general-purpose associative processor can solve this problem in about 60 milliseconds per second of real time.

Beam Splitting

Beam splitting is performed by recording the azimuth of leading and trailing edges of the target through the above process. The sum of two azimuths divided by two is compensated by subtracting the antenna beamwidth. The radar coordinates are converted to system coordinates and are transferred to the next operation.

Scan-to-Scan Association

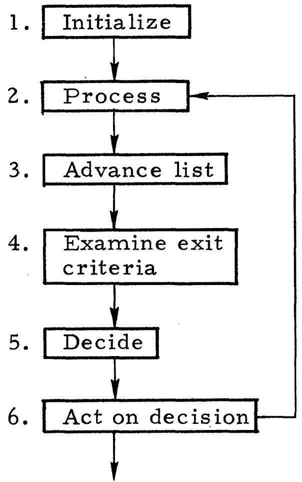
The x, y coordinates of each target report are compared, within a set of defined limits, with the predicted position of each of the tracks previously established. When association is found, the new position is recorded with the track. When more than one track associates, a minimum distance criterion is applied to establish association. If association does not take place for an input report, a new track is assumed. Beam splitting and scan-to-scan association of data take an average of 48 microseconds per report. Thus, 2000 reports per scan would require about 24 milliseconds per second.

Track Smoothing and Prediction

Periodically during the scan (say, four times), the new reports are used to smooth the track position. The new velocity is established and is used to provide a new predicted position. Those tracks that had no association during the last interval are updated on the basis of previous velocity estimates. The consistency of the track is used to develop a track quality value. Track quality is used to establish the coefficients used in the smoothing operation.

Tracks in which quality has not achieved a certain level are dropped. Tracks in which a prescribed level has been exceeded are "coasted" (that is, retained for a period of time before being dropped) a single instruction simultaneously on multiple streams of data eliminates the requirement for the familiar loops used in the conventional processor. Since the associative processor executes its instruction on all data sets at once, the loop is reduced to a single pass, and overhead bookkeeping is reduced to zero.

Conventional programming of a loop involves the following steps:

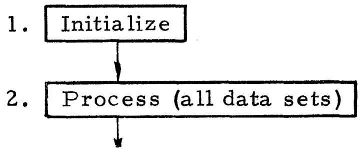


Conventional processing requires sequential execution of the complete loop for each set of data. Steps 3, 4, 5, and 6 exist once as coding overhead and "n" times as execution overhead.

To minimize the execution time, the loop should be programmed as efficiently as possible. To do this requires the use of an assembly language closely related to machine language and makes the programmer's job more difficult. That is, he must write and debug more instructions than he would for a higher level compiler language. The compiler, on the other hand, will not generate an efficient machine code for the loop, and the execution time will increase.

For the associative processor, this tradeoff between programming complexity and program execution time does not have to be made. The programming is simpler since the overhead programming is not required and since the execution time is independent of the number of data sets processed.

The equivalent flow chart for the associative processor is:



The overhead coding of the conventional loop is nonexistent, and the execution time is independent of the number of data sets processed.

Figure 5 shows that loop execution time for the conventional processor using both the assembly language and the compiler language increases with the number of data sets processed. The rate of increase is dependent on programming efficiency. It also shows that the execution time for the associative processor is independent of the number of data sets processed.

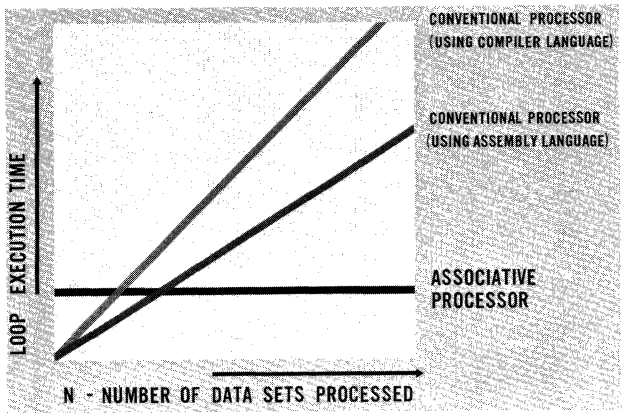


Figure 5 - Execution Time Comparison

The above example illustrates the point that the associative processor instructions are in themselves a high-level language. They combine the ease of programming features of compiler-type language and the efficiency feature of assembly language.

The fact that both programming and execution time are independent of the number of data sets also enhances the modularity concept of the associative processor. That is, the capacity of the associative processor can be expanded by adding modules without reprogramming to handle the additional data sets. This feature is extremely important in applications where the processing load is projected to increase over some reasonable period of time.

Table III lists representative sets of search and arithmetic instructions for the associative processor. The timing equations are for a particular associative processor and a particular set of microprograms, but they are typical. The equations show that the execution time is proportional to the number of bits in the fields but is independent of the number of words in a field.

APPLICATIONS

While the associative processor cannot be applied efficiently to all types of problems, there are certain types of problems for which the associative processor is clearly superior. The associative processor is ideally suited to applications that have (1) a number of data streams that can be processed in parallel and (2) operational requirements for extremely rapid data processing. Most command and control systems satisfy these criteria and, for that reason, various military organizations have been interested in the early development of associative processors.

Radar data processing for scanning radars is an interesting application for associative processors.

to see if they will recur. The track smoothing and prediction algorithm is the familiar α, β tracker.[3]

Since all tracks are treated simultaneously, the solution time is independent of the number of tracks. A significant result of this is the fact that many noise tracks can be managed efficiently. This means that a higher false alarm rate can be accepted from the detection process, and a higher probability of target detection is achieved. The effect can be equal to doubling the radar transmit power.

In conventional processing systems familiar to the authors, a high-speed processor may saturate with 300 or 400 tracks. With the associative processor, 2000 to 8000 tracks can be managed easily. The processing time is a constant and, for the algorithm described, requires 10 milliseconds per second of real time.

Display Processing

Display processing necessitates satisfying a variety of different requests. These requests require that the file be searched for tracks satisfying certain parameters. The search parameters may be area, velocity, direction, altitude, specifically assigned tracks, or identity. Limits may be assigned any of the parameters. History for the displayed tracks may be requested, and map data may be superimposed on the display. Velocity, altitude, and identity may be requested in alphanumeric form.

Finally, the data for each display must be transformed into display coordinates including the scale factor chosen and the offset of the display from the system coordinate base. The processing time for each display (assuming request changes are satisfied within one second) averages less than 1 millisecond. Thus, 50 displays would require 50 milliseconds per second of real time.

Other System Processing

Since less than 15 percent of real time is used in processing the above, a number of other system functions may be readily accommodated in the associative processor. For further depth in associative processor application studies, see References 4, 5, and 6.

Table IV lists many of the typical surveillance and control applications for the associative processor.

TABLE IV - ASSOCIATIVE PROCESSOR APPLICATIONS

	Systems[*]			
Processing functions	SC	EW	IR	C
Radar in-beam	×			
Beacon in-beam	×			
Track while scan	×			
Radar energy management (phased array radars)	×			
ECM strobe tracking	×	×		
Air traffic control	×			
Conflict prediction	×			
Threat evaluation	×		×	
Weapons direction	×			
EW pulse analysis		×		
EW threat alarm		×		
Jammer control		×		
Deceptive CM control		×		
Receiver scan control		×		
Voice pattern recognition		×		×
ELINT intercept		×		
Display processing	×	×	×	
Data retrieval	×		×	
Multilist correlation	×		×	×
Data base updating	×		×	
Message routing				×
Formatting				×
Partial data recognition			×	×
Bandwidth compression				×
Disc scheduling	×		×	
Voice encryption				×

[*] SC, surveillance and control; EW, electronic warfare; IR, information retrieval; and C, communications.

ASSOCIATIVE PROCESSOR DEMONSTRATION MODEL

The basic building block of the associative processor is the plated-wire array module. Figure 7 shows an array module that was built and demonstrated[*] in July 1969. This module contains 256 words × 256 bits of plated wire with electronics for 128 words × 128 bits. Figure 8 shows the tunnel structure/interrogate assembly that holds the 256 plated wires. Figure 9 shows the strap driver electronics with 4 drivers per pack and the wire driver electronics with 2 drivers per pack. Both are hybrids using monolithic circuits, discrete transistors, and thick film passive components.

To test and demonstrate the associative processor array module under conditions simulating system operation, a program controlled exerciser was constructed. Figure 10 shows the array module installed in a test position of the exerciser. The exerciser contains a small control memory that is used to store microprograms inserted from paper tape. Once loaded, this control memory simulates the operation of the associative processor control

---

[*] At Goodyear Aerospace Corporation, Akron, Ohio, under Contract F33-615-68-C-1555 for the Air Force Materials Laboratory, Wright-Patterson Air Force Base, Dayton, Ohio.

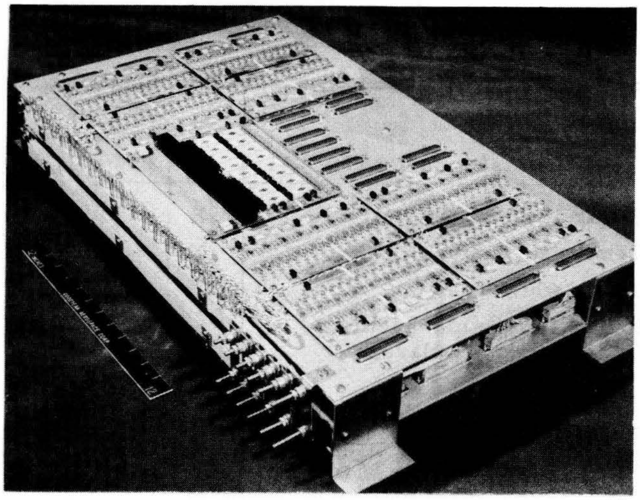


Figure 7 - Plated-Wire Array Module

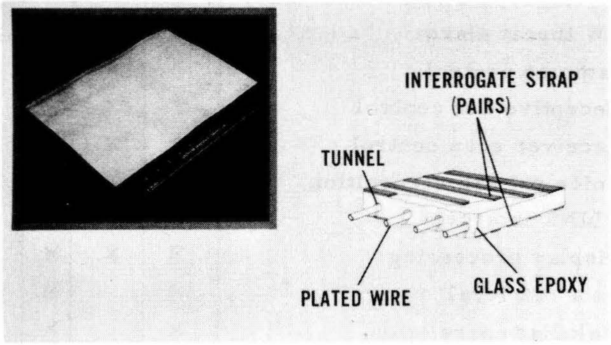


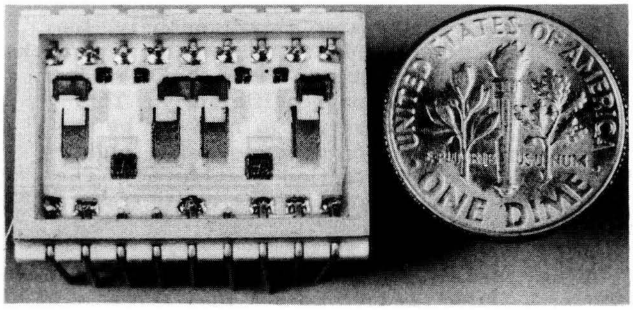
Figure 8 - Tunnel Structure

unit and can issue microprogrammed commands to the array module every 50 nanoseconds.

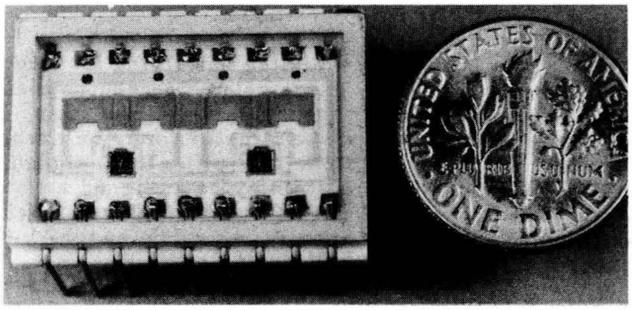
The output cabinet (far right in Figure 10) contains two displays used to measure the performance of the module. One of these is a storage tube used in conjunction with a display subroutine to display the entire contents of memory. This dynamic visual aid allows one to view the results of processing in memory. The second display is a counter that measures and accumulates the time that the module is actually active in executing the microprograms. This counter does not accumulate the display time or tape input time and, therefore, measures only associative processing time.

TEST RESULTS

The display units of the exerciser were used to demonstrate that the associative processor array module executed the instructions commanded by the control memory within the allowable times.



(A) STRAP DRIVER



(B) WIRE DRIVER

Figure 9 - Plated-Wire Array Driver Electronics

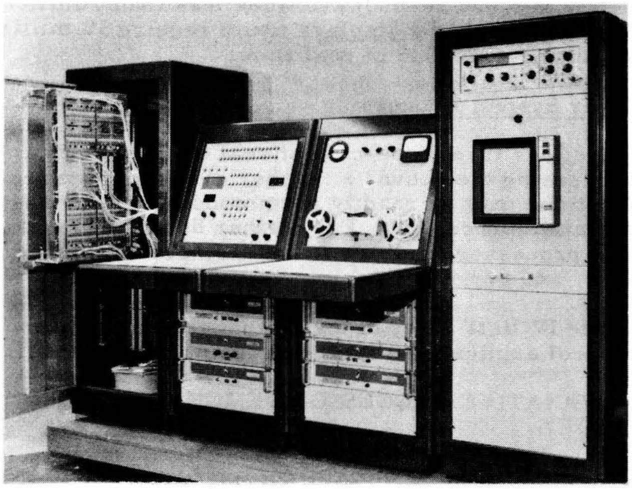


Figure 10 - Associative Processor Array Module and Exerciser

Figure 11 is a photograph of the exerciser display showing the contents of the associative processor memory at the conclusion of the test. The test proceeded as follows: Field A was loaded so that each word contained its own address. Then, a series of associative searches was executed and the results written into selected tag fields. These tag fields are marked GEC, GTC, LEC, LTC, EMC, MMC, NLC, NHC, and BLC on Figure 11.

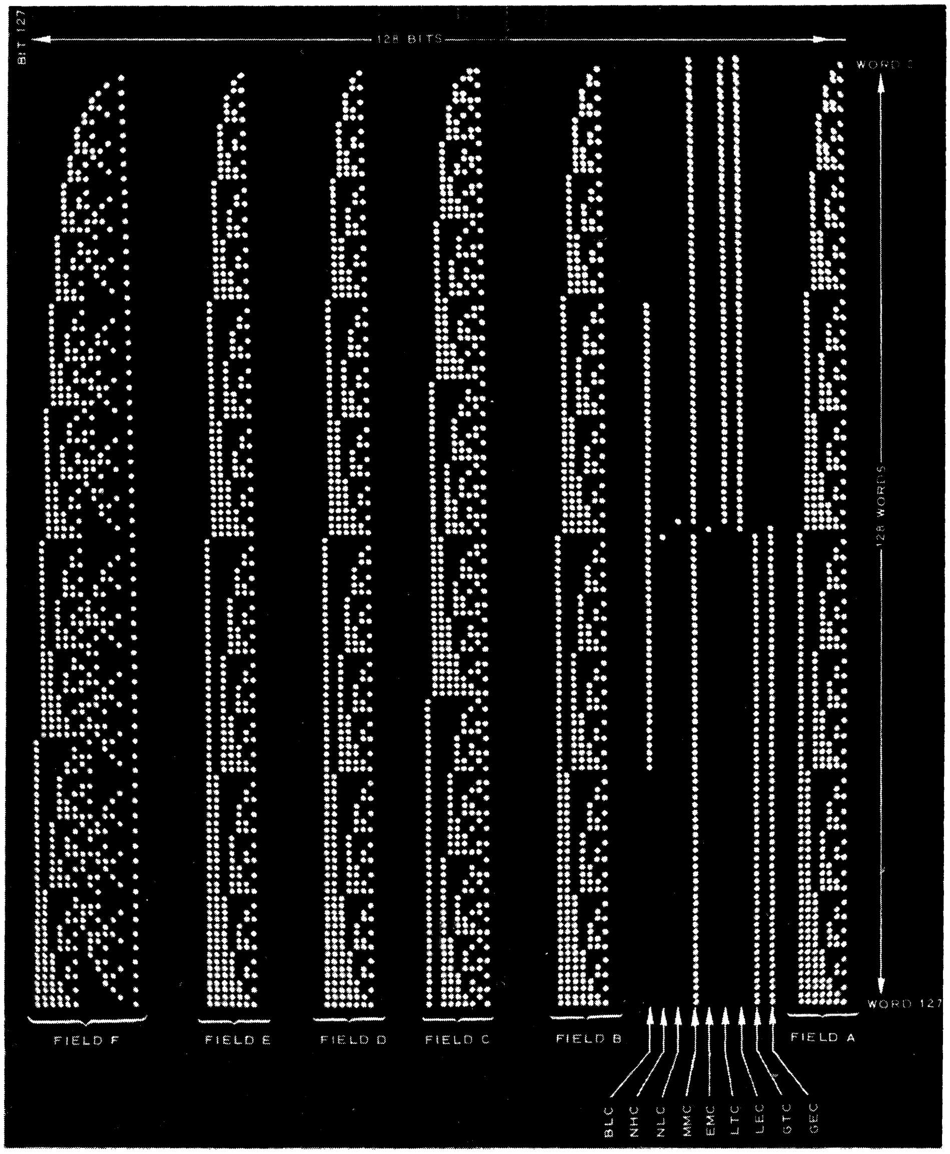


Figure 11 - Test Results Showing Contents of Associative Processor Module

These tags correspond to greater than or equal to, exact match, mismatch, next lower than, next higher than, and between limiting comparands.

Next, Field B was loaded so that each word contained its own address. Then, Field B was added to Field C (initially zero) three times. Figure 11 shows that Field C contains three times Field B.

Finally, both Fields D and E were loaded so that each word contained its own address. Then, Field D was multiplied by Field E, and the results were stored in Field F. Figure 11 shows that Field F contained the square of either Field D or E.

Table V shows the allowable and measured times for the associative processor to execute the above sequence of instructions. The allowable times were based on the timing equations derived from Table III, which in turn require the associative processor to operate at 100 nanoseconds for a parallel interrogation and 300 nanoseconds to write one bit of all words.

The actual time was significantly lower than the allowable time for the add fields and multiply fields instructions. This was because new sets of microprograms, more efficient than those represented by the allowable time, were used for these instructions.

This demonstration shows that even a small associative processor of 128 words is extremely powerful. Extending the demonstrated 10-bit add time to a 32-bit add time shows a processing rate of 4 million instructions per second for this model. Slight changes in the design of the response store will double this processing rate.

TABLE V - TIMING RESULTS*

Mnemonic operation	Definition of operation	Timing equation	Bits in field (n)	Allowable time (micro-seconds)	Measured time (micro-seconds)
GEC, TAG	Find all values in A greater than or equal to 100_8; store results	$0.1(n + 4) + 0.3$	7	1.4	1.4
GTC, TAG	Find all values in A greater than 100_8; store results	$0.1(n + 4) + 0.3$	7	1.4	1.4
LEC, TAG	Find all values in A less than or equal to 100_8; store results	$0.1(n + 4) + 0.3$	7	1.4	1.4
LTC, TAG	Find all values in A less than 100_8; store results	$0.1(n + 4) + 0.3$	7	1.4	1.4
EMC, TAG	Find all values in A equal to 100_8; store results	$0.1(n + 4) + 0.3$	7	1.4	1.4
MMC, TAG	Find all values in A unequal to 100_8; store results	$0.1(n + 4) + 0.3$	7	1.4	1.4
NLC, TAG	Find the maximum of values in A less than 100_8; store result	$0.2(n + 2) + 0.3$	7	2.1	2.1
NHC, TAG	Find the minimum of values in A greater than 100_8; store result	$0.2(n + 2) + 0.3$	7	2.1	2.1
BLC, TAG	Find the values in A that are between 40_8 and 140_8; store result	$0.2(n + 4) + 0.6$	7	2.8	2.8
ADF	Add Field B to Field C and store the results in Field C; repeat 3 times and accumulate result in C	$3[1.1(n + 1) + 0.1]$	10	36.6	30.3
MPF	Multiply Field D by Field E and store results in Field F	$1.1b(a + 2.7)$	$b = 7$ $a = 8$	82.4	63.4

*n, b, a = number of bits in field.

APPLICATION DEMONSTRATION

The associative processor array module and the exerciser also were used to demonstrate the application of an associative processor to a real-time air traffic control problem. For this demonstration, the associative processor provides aircraft motion simulation and solved the collision detection program for 128 aircraft.

Figure 12 shows a simulated airspace 64 miles by 64 miles in area containing 128 aircraft with random flight paths and velocities. Many near misses and collisions developed during this scenario.

The associative processor was programmed with a collision detection algorithm in which a 32-second look-ahead track intersection prediction is made for each aircraft against all 128 aircraft tracks. Then, only those tracks predicted to be involved in a potential collision are displayed (see Figure 13). A manage by exception display of this sort would allow the controller to concentrate on those aircraft requiring immediate attention.

The associative processor requires 3 milliseconds of each simulated 4-second radar scan to solve this collision detection problem for 128 aircraft.

SUMMARY

An associative processor, which combines the content addressability of an associative memory with arithmetic and input-output capability at each word of memory, has been described.

The associative processor is classified as a modularly expandable array processor with processing rates from 15 million to 1 billion instructions per second and input-output rates from 1 to 160 billion bits per second. Since it is constructed of inexpensive memory modules and has processing rates in excess of the supercomputers, it is projected to be 30 times as cost effective as machines in the ILLIAC IV class.

Due to its content addressability and parallel instruction execution capability, programming the associative processor is postulated to be even simpler than programming a conventional processor.

Finally, a 128-word associative processor demonstrator model has been described. The test results show that even this small associative processor is capable of operating at 4 million instructions per second.

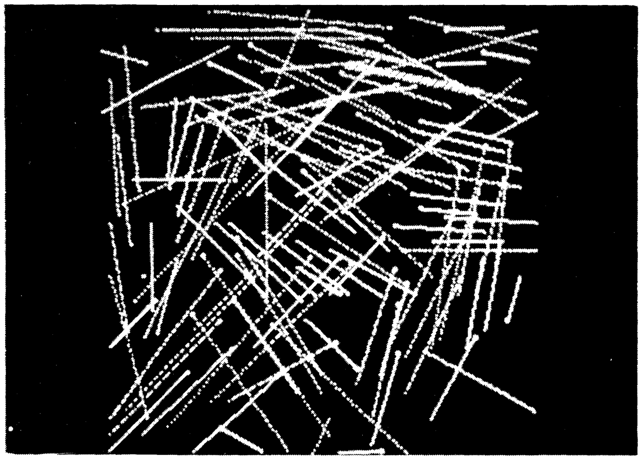


Figure 12 - Airspace Display without Conflict Detection

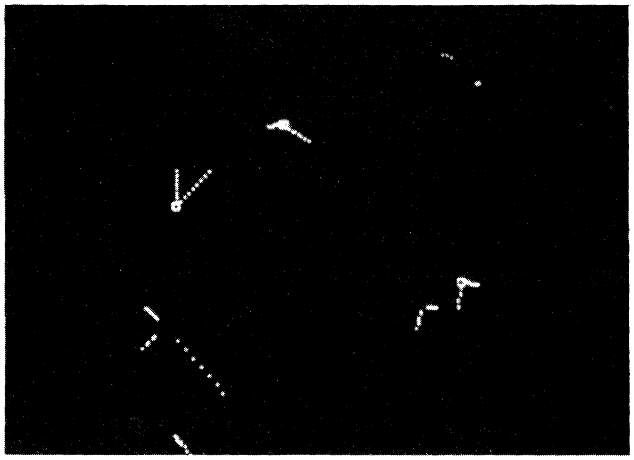


Figure 13 - Airspace Display with Conflict Detection by the Associative Processor

Meeting the demand for higher processing rates at lower costs is one of the challenges of the 1970's. The associative processor described in this paper is ready now to meet that challenge.

REFERENCES

1. Flynn, M. J.: "Very High Speed Computing Systems," Proceedings of the IEEE, Vol 54, No. 12, pp 1901 to 1909, December 1966.

2. Rudolph, J. A.: "The Associative Processor - A New Computer Resource," Proceedings, IEEE Region 6, April 1969.

3. Benedict and Bardner: "Synthesis of an Optimal Set of Radar Track-While-Scan Smoothing Equations," IRE Transactions on Automatic Control, July 1962.

4. Eddey, E. E.: "The Use of Associative Processors in Radar Tracking and Correlation," NAECON Proceedings, 1967.

5. Meilander, W. C.: "The Associative Processor in Aircraft Collision Prediction," NAECON Proceedings, 1968.

6. Costanzo, A., and Garrett, J.: "Application of Associative Processors in an Intercept Radar System," NAECON Proceedings, 1969.

AN ASSOCIATIVE MEMORY FOR EXECUTIVE CONTROL FUNCTIONS IN AN ADVANCED AVIONICS COMPUTER SYSTEM

Robert O. Berg, Melvin D. Johnson

Honeywell Systems and Research Center
Roseville, Minnesota

ABSTRACT

The paper* describes methods of using an associative memory for the implementation of executive control functions in an advanced avionics digital computer system. The various fields required in the associative memory word and functions of those fields in the task assignment routine are presented. It is shown that an associative memory can be used to mechanize an effective task assignment algorithm based on dynamic priority.

I. INTRODUCTION

Associative memories with the capability of performing search or processing operations simultaneously over all stored words are now becoming an economic reality.[2,3,4,5] This paper shows how these associative capabilities can be used to implement the tasks of an executive control unit in an advanced avionics digital computer system.

The system model (Figure 1) upon which this work

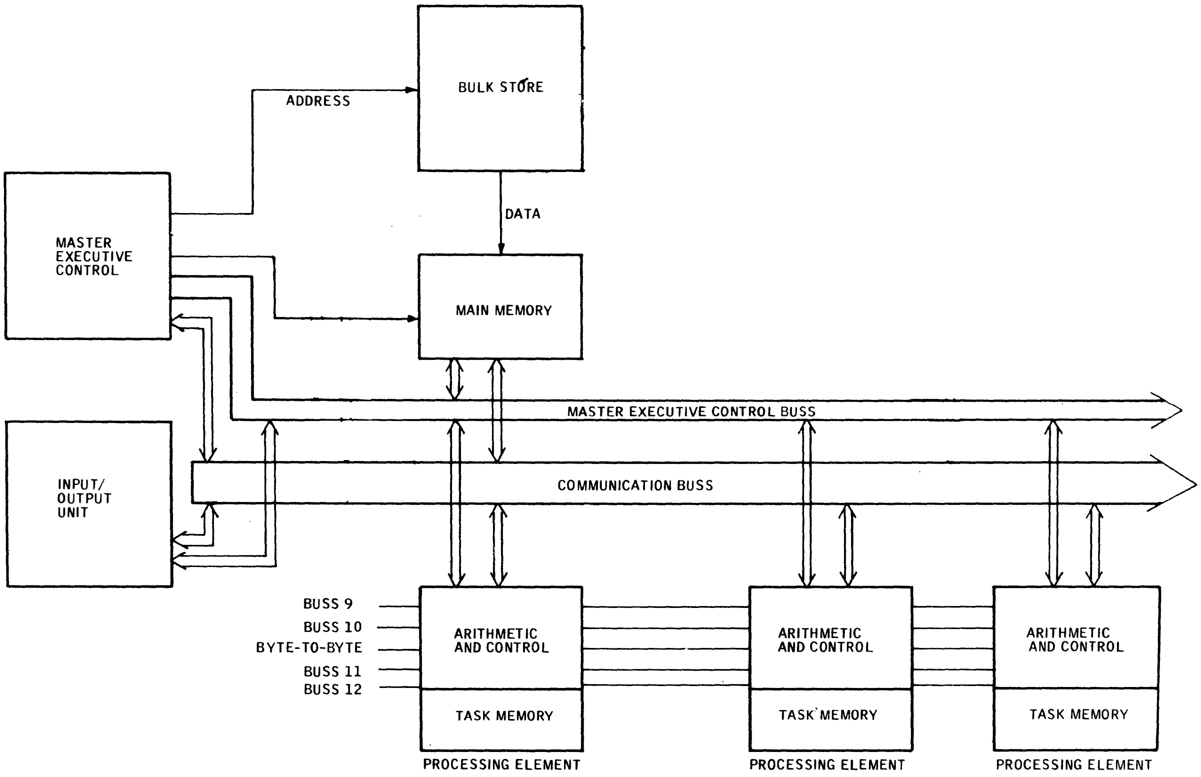

ADDRESS
BULK STORE
DATA
MASTER EXECUTIVE CONTROL
MAIN MEMORY
MASTER EXECUTIVE CONTROL BUSS
INPUT/ OUTPUT UNIT
COMMUNICATION BUSS
BUSS 9
BUSS 10
BYTE-TO-BYTE
BUSS 11
BUSS 12
ARITHMETIC AND CONTROL
TASK MEMORY
PROCESSING ELEMENT
ARITHMETIC AND CONTROL
TASK MEMORY
PROCESSING ELEMENT
ARITHMETIC AND CONTROL
TASK MEMORY
PROCESSING ELEMENT

Figure 1. Advanced Avionics Digital Computer Block Diagram

is based is defined in "Advanced Avionic Digital Computer Baseline Definition", AIR-5333F4[6] of the Naval Air Systems Command, 23 July 1969. A brief description of the system model is given first, followed by a definition of the fields required in the associative memory word and a description of their use in satisfying executive control requirements.

---

* This paper is extracted from a report entitled, "Master Executive Control Techniques for an Advanced Avionics Digital Computer System",[1] prepared by Honeywell for the Naval Air Systems Command under Project No. AIR-5333F4-69-I.

II. DESCRIPTION OF THE SYSTEM MODEL

The Advanced Avionics Digital Computer (AADC) system used as a model in this study consists of the following major elements:

- Bulk Store -- A mass memory unit, designed to store all operational programs for the computer system.
- Main Memory -- A random access memory, used for temporary storage of programs prior to execution.
- Processing Elements -- General-purpose processors, capable of the logic and arithmetic operations necessary for handling sequentially organized computational tasks in the AADC (such as navigation, weapons inventory, etc.). The AADC contains a number of these processing elements, with the exact number variable.
- Task Memories -- High-speed (probably semiconductor) memories used for storage of instructions and data while they are being executed by a processing element.
- Input/Output Unit -- A sampling and storage unit used for transfer of data between the AADC and the remainder of the avionics system.
- Executive Control Unit (ECU) -- A program memory, an associative memory, and a logic and control section. This unit is responsible for the following functions:
  1) Continuously monitoring the various processing elements (PE's) in the system to meet the requirements of all (externally-requested) modes of operation of the aircraft.
  2) Supervising the collection and transfer of input/output data within the AADC.
  3) Assigning operational programs to the various processing units.
  4) Supervising data transfers between the other units within the AADC, such as the transfer of programs between bulk store and main memory.
  5) Supervising the overall system operation for such items as processor failures, interrupt requests, etc.

The associative memory of the ECU must be capable of performing at least the following associative functions:

- Increment
- Decrement
- Bit slice write
- Masked equality search
- Maximum search
- Minimum search
- Less than or equal to search

These requirements will become obvious in the description of the "Assignment Function."

This particular model, which represents only one of a number of advanced concepts being considered by the Navy, exhibits three distinguishing features:

- Use of a separate and distinct hardware unit for all executive control functions. This ECU must supervise all job assignments, data transfers, and overall systems operation.
- The use of modular programs. The overall processing requirements of the system are separated into a number of modes of operation. Each mode then consists of a number of computational tasks called program modules (PM's). Thus the proper operation of the system requires the computation of several PM's at a given rate and in the proper sequence to effect completion of all required tasks in a given mode of operation.
- Dynamic configuration of variable-sized processors, from a number of byte-sized processing elements, for the execution of modular programs.

The combination of these features should result in a very high throughput multi-processing system in which most of the shortcomings of conventional organizations have been eliminated.

III. SCOPE OF CONSIDERATION

While a number of Executive Control functions was considered in the study, this paper deals primarily with the "Assignment Function," that is, the manner in which the associative memory in the ECU is utilized to establish dynamic priorities between program modules and ensure that PM's are assigned for processing in a proper order.

The relationship of this function to the total Executive Control function is shown in the flow chart, Figure 2. Thus, in the discussion which follows, it is assumed that:

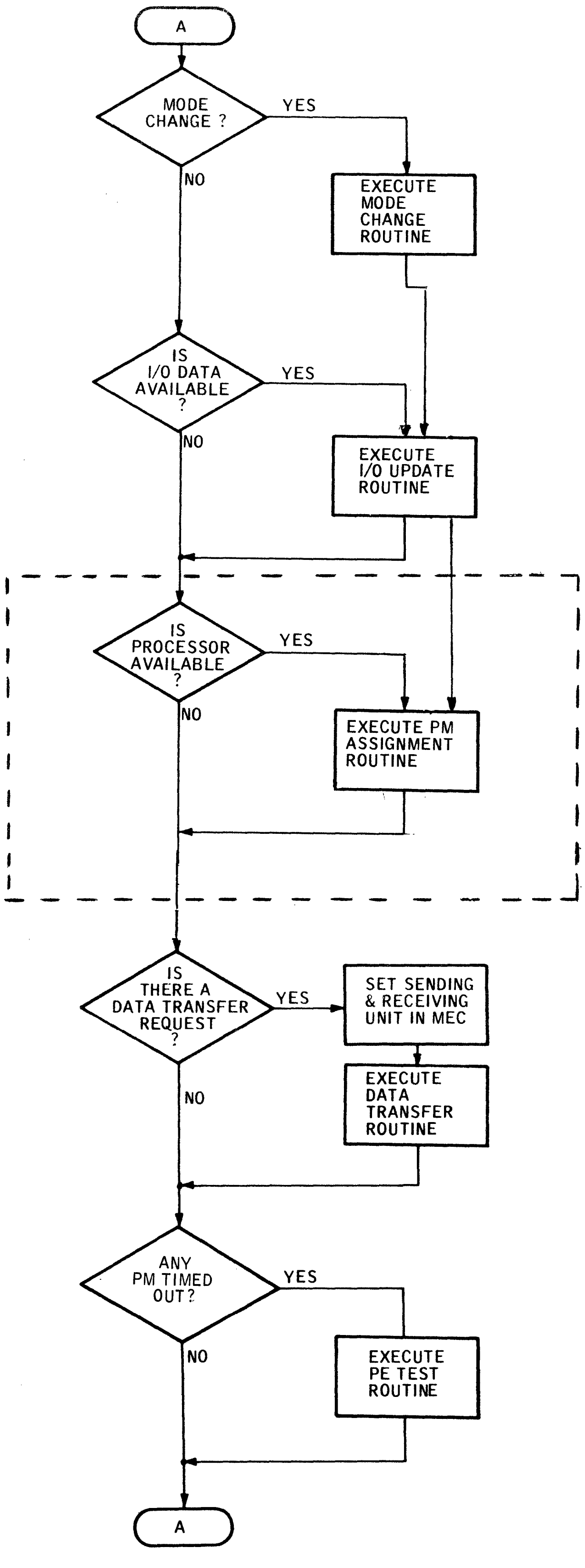

A
MODE CHANGE ?
YES
NO
EXECUTE MODE CHANGE ROUTINE
IS I/O DATA AVAILABLE ?
YES
NO
EXECUTE I/O UPDATE ROUTINE
IS PROCESSOR AVAILABLE ?
YES
NO
EXECUTE PM ASSIGNMENT ROUTINE
IS THERE A DATA TRANSFER REQUEST ?
YES
NO
SET SENDING & RECEIVING UNIT IN MEC
EXECUTE DATA TRANSFER ROUTINE
ANY PM TIMED OUT?
YES
NO
EXECUTE PE TEST ROUTINE
A

Figure 2. Executive Control Unit Main Timing Loop

1) Any mode change has been recognized, and all necessary PM's are resident in main memory.

2) All I/O updating has been completed.

IV. DESCRIPTION OF THE ASSIGNMENT FUNCTION

In order to control the assignment of program modules to processing elements, each PM has an Identification (ID) word associated with it. The ID words for all PM's in the current mode are then maintained in the associative memory of the ECU. Thus, in order to completely describe the assignment function it is necessary to consider:

1) The contents, or "fields" of the various ID words

2) How these fields are used to execute the assignment function.

Both of these items are discussed for each "step" of the assignment function. The entire PM assignment routine is shown in Figure 3.

As shown in Figure 3, there are two ways to enter the PM assignment routine from the mode change routine or, after the regular updating of input data, from the main timing loop. Assuming that entry has been made from the latter point, and a processor has indicated that it is available, the first task (Figure 3) is to identify which PM has just been completed by that processor. Two fields of the ID words are utilized in this operation:

Field 1 -- Program Module Number -- This field contains a unique code, or name, for each program module.

Field 2 -- Identity of Processor Assigned -- As each PM is assigned for processing, this field is loaded, by the ECU, with a unique code identifying the processor to which it is assigned.

With the above fields, and assuming that an available processor identifies itself by name, a single search over Field 2 will result in a match for only one ID word. A read-out of Field 1 for that word will then identify the PM assigned to that processor.

Having identified the PM which has just been completed, the next task of the ECU is to "notify" any other PM's that might utilize this one as a "predecessor."

In order for the ECU to maintain this interdependence of PM's, a third field was defined for each ID word:

Field 3 -- Predecessor Field -- This field consists of a number of bits equal to the total number of PM's currently active in the system. Each bit position then corresponds to a PM number. A "1" in any bit position indicates

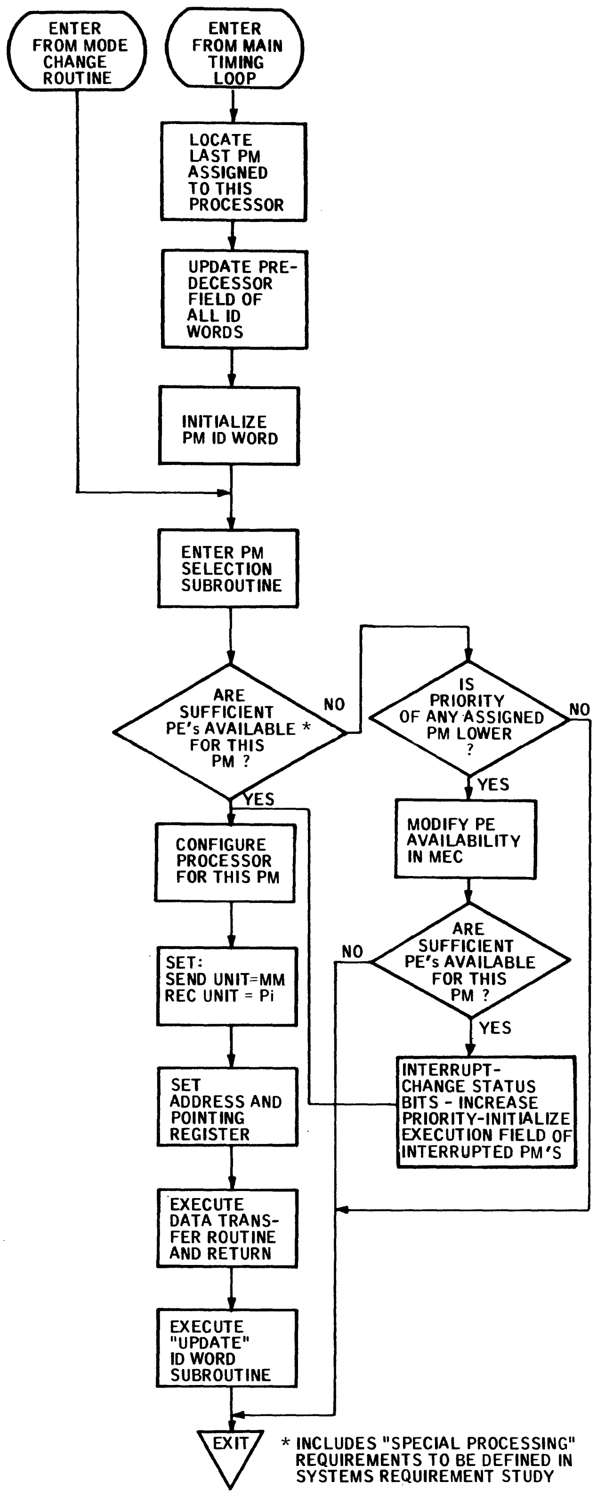

ENTER FROM MODE CHANGE ROUTINE
ENTER FROM MAIN TIMING LOOP
LOCATE LAST PM ASSIGNED TO THIS PROCESSOR
UPDATE PREDECESSOR FIELD OF ALL ID WORDS
INITIALIZE PM ID WORD
ENTER PM SELECTION SUBROUTINE
ARE SUFFICIENT PE's AVAILABLE * FOR THIS PM ?
NO
YES
IS PRIORITY OF ANY ASSIGNED PM LOWER ?
NO
YES
MODIFY PE AVAILABILITY IN MEC
ARE SUFFICIENT PE's AVAILABLE FOR THIS PM ?
NO
YES
INTERRUPT-CHANGE STATUS BITS - INCREASE PRIORITY-INITIALIZE EXECUTION FIELD OF INTERRUPTED PM'S
CONFIGURE PROCESSOR FOR THIS PM
SET: SEND UNIT=MM REC UNIT = Pi
SET ADDRESS AND POINTING REGISTER
EXECUTE DATA TRANSFER ROUTINE AND RETURN
EXECUTE "UPDATE" ID WORD SUBROUTINE
EXIT
* INCLUDES "SPECIAL PROCESSING" REQUIREMENTS TO BE DEFINED IN SYSTEMS REQUIREMENT STUDY

Figure 3. Program Module Assignment Routine

that the PM with that number must be completed before this PM can be assigned.

The use of the predecessor field is demonstrated in the example below, for four PM's. The required order of execution for the example is shown in the diagram of Figure 4. The contents of the predecessor field, before any of the PM's have been assigned, are shown in Figure 5.

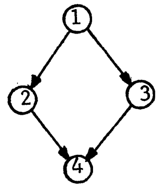


Figure 4. PM Execution Sequence

PM Nr.	Bits Corresponding To			
	PM1	PM2	PM3	PM4
1	0	0	0	0
2	1	0	0	0
3	1	0	0	0
4	1	1	1	0

Figure 5. Predecessor Field Contents

Thus, PM1 has no predecessors. PM's 2 and 3 have PM1 as a predecessor, etc. With this field, the "notification" of all other PM's that a particular PM has been completed can be accomplished by a simple "bit slice write" operation. That is, once the number of the completed PM has been located, the ECU will simply write a zero simultaneously in the proper bit position of the predecessor field of all ID words.

The only remaining function associated with the completed PM is the reinitialization of the ID word associated with that PM. Without detailing which fields of the word must be reinitialized (this will become evident later), it is sufficient to point out that the information necessary for this operation is contained in Main Memory. The ECU then merely accesses Main Memory, and initializes the ID word via a masked "write" operation.

A new PM must now be selected for assignment to the available processor. This selection process is shown in Figure 6, and involves two additional fields of the Associative Memory word.

Field 4 -- Schedule -- Initially, this field contains a digital value which represents the rate at which this PM is to be executed. This field is continuously "counted down", and its value used to adjust the priority of the PM. The schedule field will continue to be decremented when a PM is sent to a processor, in case an interrupt prevents its completion. Upon completion this field will be reinitialized.

Field 5 -- Priority -- When an ID word is first brought into the associative memory, it has a basic priority. This priority is dynamically

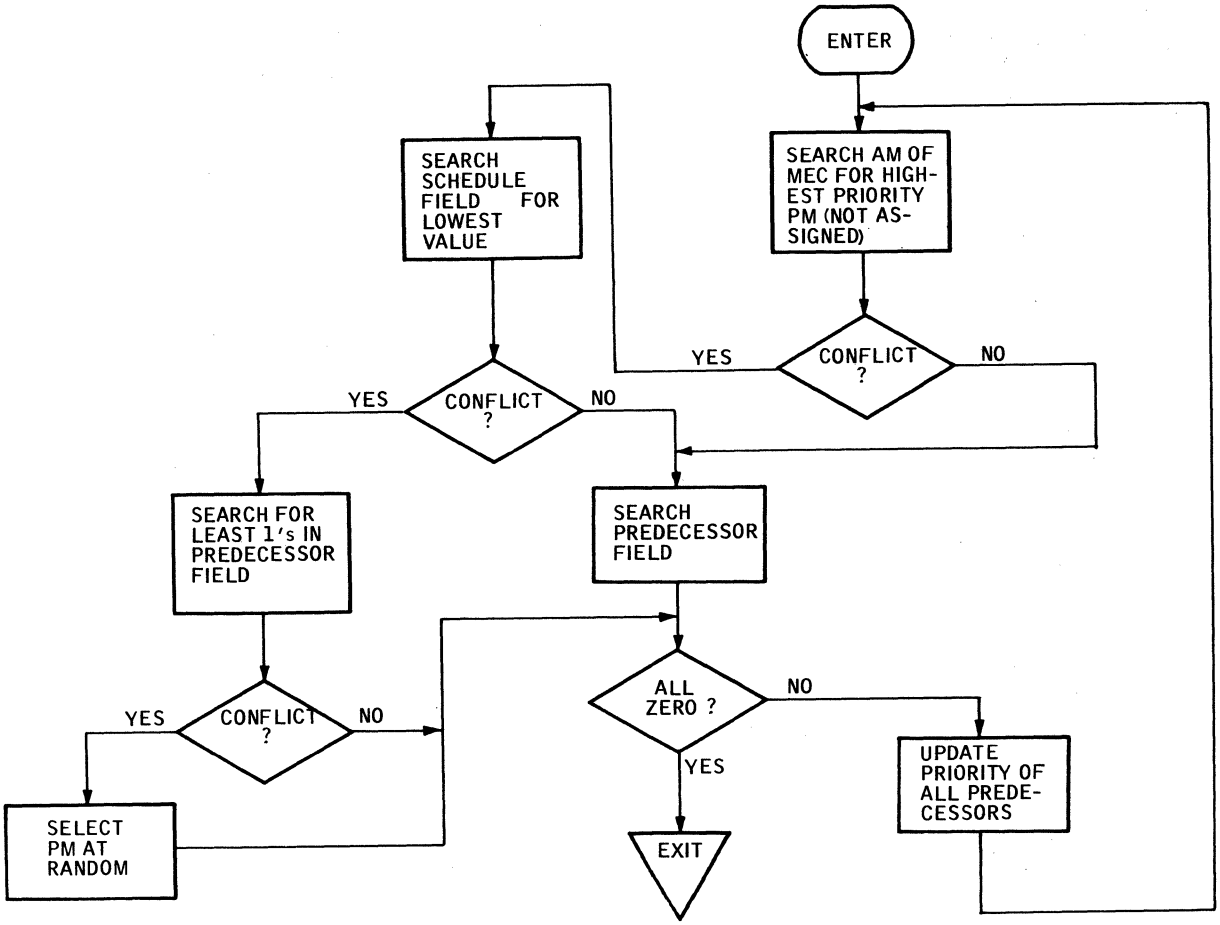

ENTER
SEARCH AM OF MEC FOR HIGHEST PRIORITY PM (NOT ASSIGNED)
CONFLICT ?
YES
NO
SEARCH SCHEDULE FIELD FOR LOWEST VALUE
CONFLICT ?
YES
NO
SEARCH FOR LEAST 1's IN PREDECESSOR FIELD
SEARCH PREDECESSOR FIELD
CONFLICT ?
YES
NO
SELECT PM AT RANDOM
ALL ZERO ?
NO
YES
UPDATE PRIORITY OF ALL PREDECESSORS
EXIT

Figure 6. Program Module Selection Subroutine

changed by ECU logic on the basis of the changing schedule and predecessor fields. When a PM is sent to a processor, the priority is biased upwards to avoid interruption by an unassigned PM which would otherwise have equal priority. This dynamic priority will govern the order of transfers of PM's to a processor. Upon completion of a PM execution, this field will be reinitialized in the associative memory.

Using the description of these fields, and the flow diagram of Figure 6, the selection process can be described as a series of search operations. Thus, the ECU first searches the associative memory priority field for the highest priority PM not assigned to a processor. If only one PM has the highest priority, it is initially accepted. If there is more than one PM with the highest priority, ECU will search these ID words for the lowest value in the schedule field. If there is no conflict, the PM with the lowest schedule field is initially accepted. If a conflict still exists, ECU performs a search for the least number of 1's in the predecessor field of the competing PM's ID words. If no conflict exists, the PM with the fewest predecessors is initially accepted. If a conflict still exists, the ECU will initially select a PM at random (i.e., lowest numbered). The initially-accepted PM is then checked to determine if the predecessor field is all zeros; if it is zero, the PM is finally accepted. If the predecessor field of a PM does not contain all zeros, the priority of all its predecessors is increased, and a search for the highest priority PM is begun again and repeated until a final selection is made.

When a PM is finally accepted, it must be determined if enough byte sized processing elements (PE's) are available to configure a processor for this PM (see Figure 3).

The fields of the ID required for this task are:

Field 6 -- Number of bytes required -- This field contains the binary representation of the size of processor (number of PE's) required to execute this PM.

Field 7 -- Which PE's make up processor -- This field contains 1 bit for each PE in the system. When a processor is assigned to a PM, a one will be written into the proper bit of this field for every PE in the processor.

This field is set to zero when the PM has been computed.

Thus, field 6 indicates the number of PE's required. A search over field 7 of all ID words for a series of adjacent zeros will be made to determine if a suitable processor can be configured. This search is successful when the first group of adjacent zeros of sufficient length is encountered.

Assuming the search is successful, the ECU will configure the processor and effect the transfer of the PM to the processor.

The final step in the assignment routine which utilizes the Associative Memory is the "Update ID Word Subroutine" shown in Figure 7.

Two additional fields of the ID word are required for this function:

Field 8 -- Execution time -- This field contains a value indicating the computation time limit for this PM. This field is counted-down during the period the PM is being executed. If this field reaches zero before the PM is completed erroneous operation is assumed and diagnostics are initiated. This field is reinitialized after PM computation.

Field 9 -- Execution status -- This field identifies the location of the PM as in Bulk Store, in Main Memory or in Processor.

Upon PM assignment the priority of its ID word is increased to prevent erroneous interrupts (see Figure 3). The execution time countdown is then begun and the status changed to "in Processor."

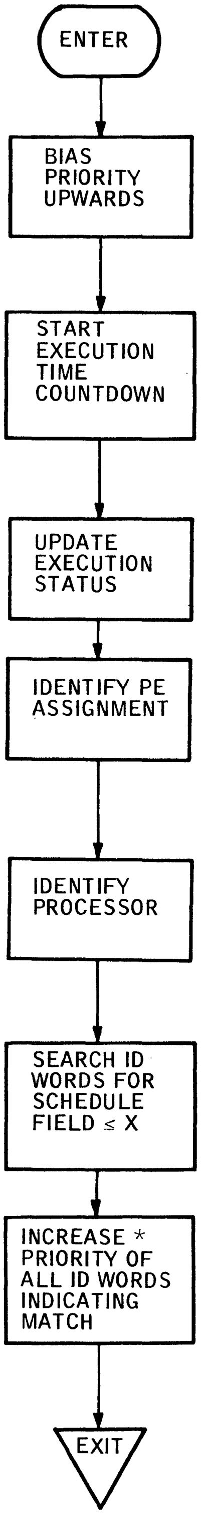
The processor and PE's assigned are then entered into the proper fields of the ID word.

A search is then made over the schedule fields of all words in the Associative Memory to locate all ID words whose schedule field has counted down to a preselected value or less. All ID words indicating a match will have their priority increased by a given value.

SUMMARY

This paper presents a method of utilizing an associative memory to implement some of the executive control function in an advanced avionics digital computer system.

In order to provide a background for the executive requirements, a brief overview of the Navy AADC system is provided. Significant features of this system include the use of modular programs and the use of a distinct hardware unit as the Executive Control Unit. The ECU must have the ability to dynamically configure variable length processors for the execution of the modular operational programs after having first determined which program is to be assigned. The heart of the ECU is an associative memory which contains identification words (ID's) to efficiently define and control the modular programs of the system.


ENTER
BIAS PRIORITY UPWARDS
START EXECUTION TIME COUNTDOWN
UPDATE EXECUTION STATUS
IDENTIFY PE ASSIGNMENT
IDENTIFY PROCESSOR
SEARCH ID WORDS FOR SCHEDULE FIELD ≤ X
INCREASE * PRIORITY OF ALL ID WORDS INDICATING MATCH
EXIT

* MORE RESOLUTION CAN BE OBTAINED BY SEARCHING ON MORE THAN ONE LIMIT.

Figure 7. Update Identification Word Subroutine

The implementation of the program module assignment function using an associative memory is presented. This implementation is described in the form of flow diagrams and field descriptions of the associative memory words. This rather elaborate assignment routine can be realized through the utilization of an associative memory capable of performing the following associative functions:

1) Increment
2) Decrement
3) Bit slice write
4) Masked equality search
5) Maximum search
6) Minimum search
7) Less than or equal to search

The assignment routine presented in this paper makes extensive use of search operations, incrementing and decrementing operations, and a selective write operation. Each of these operations can be accomplished simultaneously over all words, or a selected subset of words, in an associative memory. Use of a conventional memory to accomplish these necessary functions would require that each ID word be handled serially. This would obviously result in a much slower implementation of the assignment routine. This inherent speed advantage, coupled with the decreasing cost and increasing availability of associative memories, will soon make them very useful, if not essential, in future high-speed systems.

REFERENCES

1. Berg, R.O. and Johnson, M.D., "Master Executive Control Techniques for an Advanced Avionics Digital Computer System," Honeywell Document 14206-FR, July 1969.

2. Leonard, D.N., "MOS Memories," EEE, November 1969.

3. Herlein, R.F. and Thompson, A.V., "An Integrated Associative Memory Element," ISSCC Proceedings, 1969.

4. Jordan, W.F., "MOS Arrays Come on Strong," Electronics, 23 December 1968.

5. Wald, L.D., "An Associative Memory Using Large-Scale Integration," NAECON Conference Record, May 1970.

6. Naval Air Systems Command, "Advanced Avionic Digital Computer Baseline Definition," NASC AIR-5333F4, 23 July 1969, Washington, D.C.

SESSION SUMMARY:
ROTATING MEMORIES AND ELECTRONIC DRUMS -- PERFORMANCE AND COST CONSIDERATIONS

Edwin I. Blumenthal, Defense, Space & Special Systems Group, Burroughs Corporation

Our panelists have addressed themselves essentially to two issues:

1. The relative advantages (cost/performance and reliability) among the various types of rotating memory devices-head per track drums or disks (access times to as low as 5 ms.), removable media/moving head files (such as disk pack drives), and fixed disk/moving head files, where access times approach 100 msec; and

2. The potential replacement of (the faster) rotating memory device with solid-state "electronic drums".

In addressing the first issue, Mr. Tickell reviews the state-of-the-art with respect to all three classes of rotating memories. This sets the stage for the positions taken by Messrs Jack, Wieselman and Yang, who present the case for head per track files, fixed disk/moving head files, and disk pack drives, in that order.

The result of these first four papers is thus a summary of the inherent advantages of one over the other, providing answers to questions such as- What price higher access speeds for head per track disks or drums? What price media removability?

Table I is a rough concensus of all four panelists on the cost of storage by class of rotating memory device, and how these costs are expected to change within the next five years. This table clearly indicates the target presented to the designer of solid state "equivalents".

The second issue is then addressed. Dr. Arnold's paper proposes the criteria to be employed by the systems designer in evaluating the desirability of replacement of high speed drums or disks with new devices.

Finally, Messrs Lloyd and Broadbent present their candidates for such replacement, in the form of two quite different solid-state memory devices; one built out of MOS shift registers, the other employing plated wire.

END-USER PRICE * (¢/BIT)

	Fixed-Disks/Moving Heads	Removable Disks/Moving Heads	Head per Track/Fixed Disk or Drum		
Avg. Access	(40 to 100ms)	(40 to 100 ms)	5ms	20ms	50ms
1970	.005	.008	.2	.02	.014
1975	.0005 - .0015	.0008 - .002	.03-.05	.003 - .005	.002 - .003

---

* Session ground rules established ¢/bit to the end-user as a standard. To convert between cost bases, we make the crude assumption that end-user price is twice that of OEM (large-quantity) price, while the latter is twice the manufacturing cost.

TABLE I

STORAGE IMPROVEMENTS AND SYSTEMS' COST/PERFORMANCE

R. F. Arnold, International Business Machines Corporation

INTRODUCTION

It is unpleasant but generally recognized that the body of the existing applications and programs for storage based systems largely limits new storage products to those which are superior with respect to both cost and performance to those products which they replace.

This issue is reviewed in the context of a simple, one parameter performance equation which is consistent with much of the author's experience in this area. The equation provides a rule of thumb for extrapolating systems thruput improvement when systems components are upgraded.*

Following a description of the equation, an attempt is made to shed light on two current problems in storage planning.

1. Given a certain advancement in technology, and a linear device cost/performance curve, how should one best configure the technology to optimize system cost/performance?
2. What must the currently proposed electronic storage products cost in order to qualify as "drum replacements"?

In both cases the exercise suggests a basis for our initial suspicions. The model is offered in the spirit of a challenge to those who might want to elaborate or replace it with one of their own and a warning not to neglect the cost, performance, and usage characteristics of current storage technology in planning future products.

THE PERFORMANCE EQUATION

Here is a description of the "model".

Assume a real system with several components whose utilizations can be measured by some combination of systems measurement hardware, software, and guessing. There is then available for a given job stream a set of times, t_1, t_2, t_3 etc. which are the numbers of seconds which resources 1, 2 and 3 were busy. There is also available an overall time T which it took to complete the job stream.

These values may be substituted into the following equation and the unknown parameter S determined numerically.

$$T = (t_1^{\frac{1}{S}} + t_2^{\frac{1}{S}} + t_3^{\frac{1}{S}})^S$$

S is called the sequential dependence of the job stream. Inspection of the equation shows that for S = 1, T is equal to the sum of the resource times which represents the worst case of no resource overlap. As S approaches 0, T approaches the maximum of the resource times, which is the best case of complete overlap. In practice values of S have been observed between .25 and .75, the low values occurring in highly optimized systems, usually dedicated to a specific application whose structure is well understood.

Now if a particular resource 1 is replaced by a new 1', and no other changes made, it is very often possible to derive a new time t_1, for the new device using data like average record length, number of accesses, etc. If one assumes S remains fixed one can substitute the new t_1 back in and determine a new T, and thus a prediction of the new job stream time. I suggest that this prediction is a useful guide if the following assumptions hold:

1. There is no reallocation of data sets among devices.
2. The same scheduling and other operating system characteristics hold.
3. The same blocking and access patterns hold.

Very often assumptions such as these are warrented for the initial application of a broadly marketed storage product.

---

*For a mathematical treatment of these functions see Reference 1.

STORAGE CONFIGURATION

The above equation can be combined with cost data to examine system cost/performance. It is generally true that within a given level of technology, cost and performance can be traded off. We consider the case where this trade off is linear, an assumption which is not too unrealistic for drums in an access time sensitive environment for example.

Suppose a three-component system, each component having unit cost and taking one second to process its part of a job stream. Then the base system cost/performance will be given by:

$$C/P = \underset{\text{COST} \times}{(1+1+1)} \; \underset{\text{TIME}}{\left(1^{\frac{1}{S}} + 1^{\frac{1}{S}} + 1^{\frac{1}{S}}\right)^{S}}$$

$$= 3^{S+1}$$

which ranges from 3 to 9 depending on the sequential dependence.

Now let us assume that an improvement in the technology of the first component occurs, and that

$c_1 t_1 = A < 1$

(A = "State of the Art")

(A = 1 in the current technology)

Then one can substitute in

$$t_1 = \frac{A}{c_1}$$

and obtain system cost/performance as a function of device cost

$$C/P = (c_1 + 2)\left(\left(\frac{A}{c_1}\right)^{\frac{1}{S}} + 2\right)^{S}$$

Figure I shows this function for S = 1/2 and values of A of 1/2 and 1/4. Notice that in both cases, at the optimum point, more of the improvement is allocated to cost reduction than to performance enhancement, and that after you have paid too much for it, it does not make any difference how good it is.

Figure II shows S = .25 and S = .75 for A = 1/2. Again, cost reduction is more important than performance enhancement, but not surprisingly, the system with the most sequential dependence (S = .75) should invest a little more in performance enhancement.

In the absence of more precise inputs, the most these equations can do is remind us again that success depends on being better in both cost and performance, with a greater emphasis on cost than on performance. Taking derivatives one finds

$$C_{\text{OPTIMUM}} = A^{\frac{1}{1+S}}$$

so that for S = 1 one should split the improvement equally between cost and performance while for S = 0, all the improvement should go into cost reduction.

Operating in the opposite direction are the twin effects of the performance advance of other systems components and development of new programs which fit the new device better. The gradual evolution of these during the product life will increasingly move the optimum device configuration toward higher cost and higher performance. One is always left with the problem of assessing the rate of this advance to effect a compromise which insures enhanced system cost/performance on day one.

"DRUM" REPLACEMENT CRITERIA

It is of interest to look at some cost/performance curves which might apply to any of the non-rotating 'drum replacement' technologies. If one assumes that they must cost justify themselves only as drum replacements in the sense I have spoken of, then it is clear that they had better not cost much more than the drums they replace, no matter how fast they are. The numbers resulting from the following arbitrary numerical assumptions are not offered as absolute criteria. They do provide a representative example.

Assume the drum to be replaced represents 10% of the system cost and is one of ten equally used system components. Also assume a sequentialness of 1/2. Then a system having unit cost and unit performance would obtain if for each component we had

$$\text{Cost}_i = .1 \quad t_i = \sqrt{.1} = .31$$

Notice that the drum usage is about 31%. (This would include latency as well as actual data transfer.)

Now let us assume the replacement of the drum by a device which reduces the drum time to zero and see how the cost/performance of the system is affected as a function of the cost of this infinitely fast drum (I.F.D.).

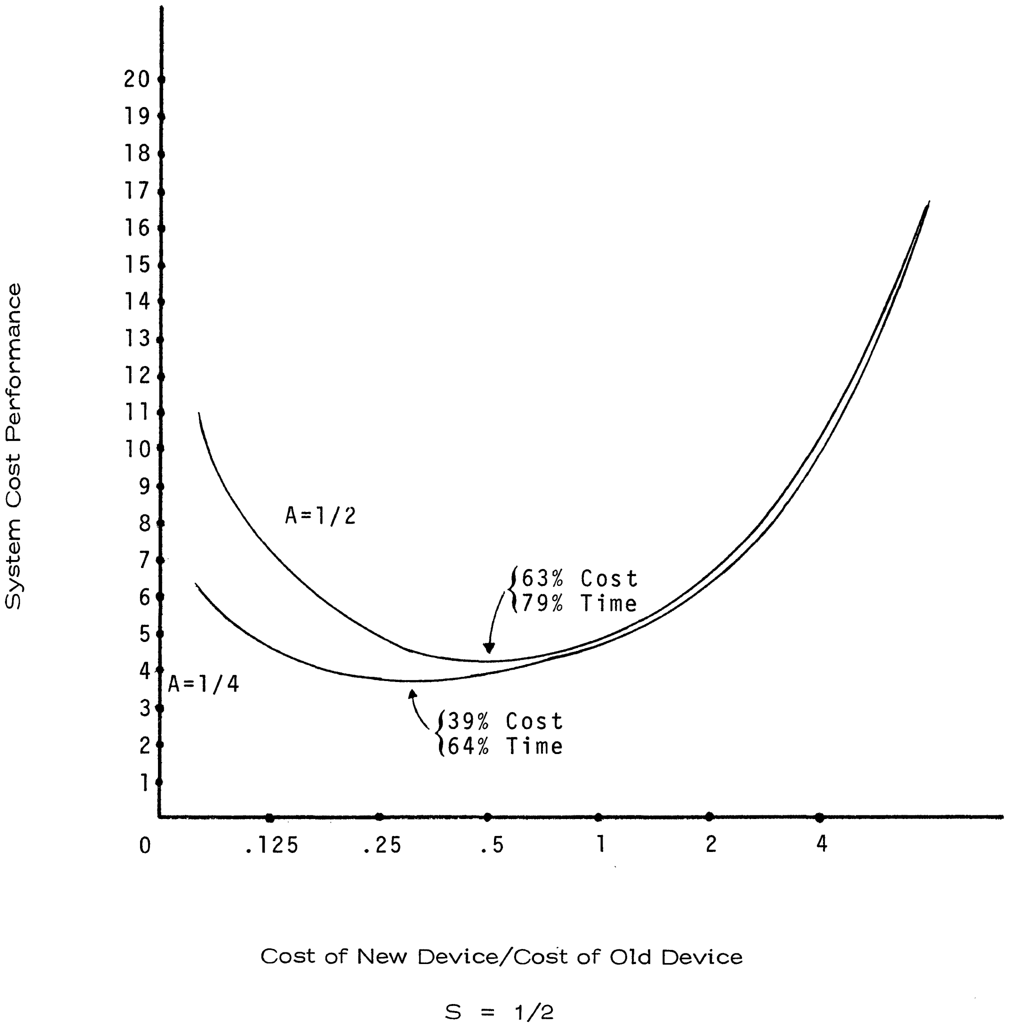

System Cost Performance
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
.125
.25
.5
1
2
4
A=1/2
A=1/4
63% Cost
79% Time
39% Cost
64% Time
Cost of New Device/Cost of Old Device
S = 1/2

FIGURE I

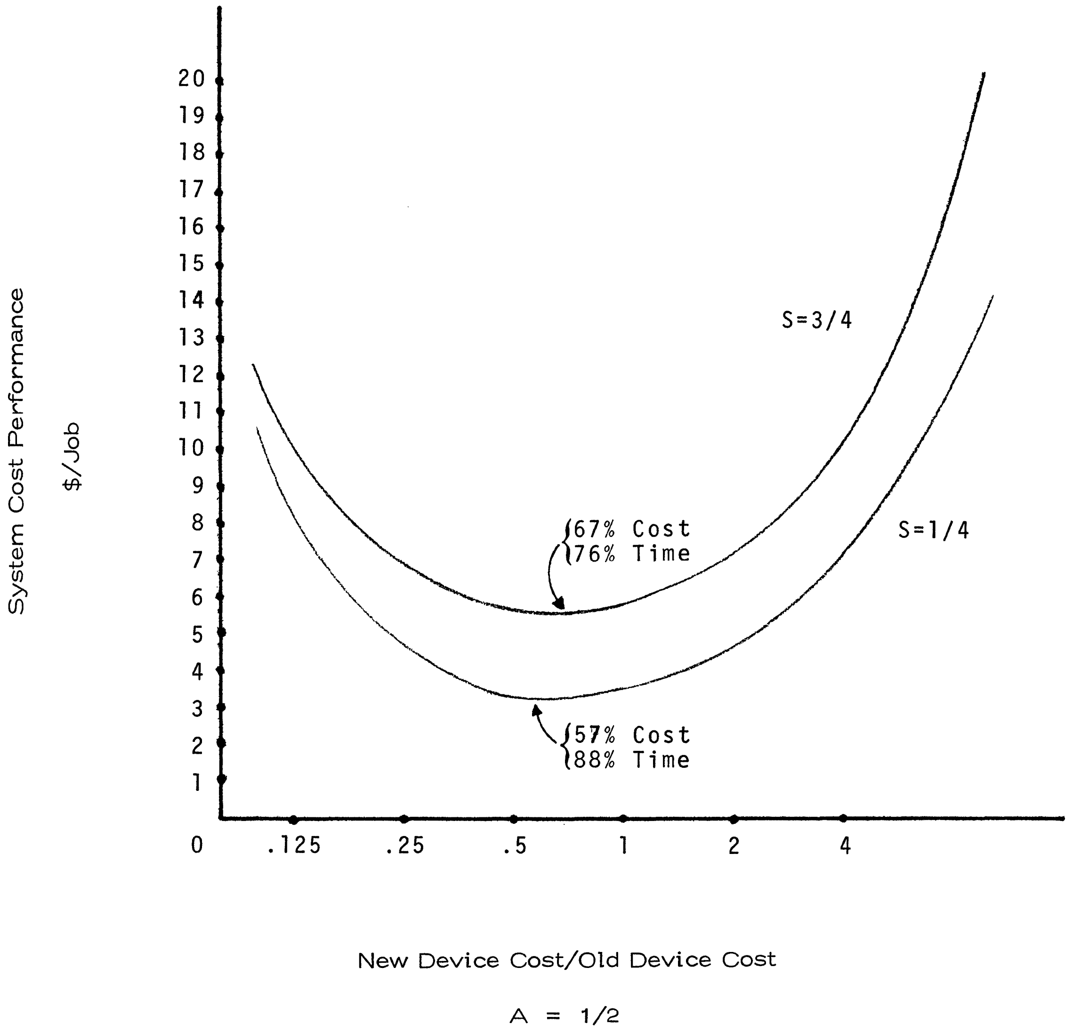
System Cost Performance
$/Job
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
.125
.25
.5
1
2
4
S=3/4
S=1/4
67% Cost
76% Time
57% Cost
88% Time
New Device Cost/Old Device Cost
A = 1/2

FIGURE II

Cost of the new system is

$$(.1C_{IFD} + .9)$$

and job time is

$$(0 + .1 + .1 + .1 + .1)^{\frac{1}{2}} = (.9)^{\frac{1}{2}}$$

Where C_{IFD} is the cost of the IFD expressed as a multiple of the old drum cost. Then relative cost/performance of the systems with the IFD is

$$\begin{aligned} C/P_{IFD} &= (.1C_{IFD} + .9)\,(.9)^{\frac{1}{2}} \\ &= (3C_{IFD} + 27)/10\sqrt{10} \end{aligned}$$

This is shown on Figure III.

Two interesting points on this curve are noted. $C_{IFD} = 1$ represents equal device cost and shows a cost/performance improvement of about 5%. $C/P = 1$ represents equal cost/performance which occurs at a device cost 50% greater than that of the drum. If the drum price is .2¢/bit, then the IFD is a break even proposition only at .3¢/bit.

Also shown on the figure is the line for a finite improvement factor of 2. This already achieves 75% of the ultimate improvement potential and an improvement factor of 10 achieves about 99% of this potential.

One should reasonably ask what would have happened if the other system components were improved? Might this not cause the drum time to "stick out" and thus justify a greater expenditure on the drum replacement?

Line B on Figure IV shows this effect assuming a 4 times improvement in all other systems components performance, and does indeed suggest that one could get by with a higher IFD cost. However this apparent opportunity is largely negated by the fact that the other systems components will tend to experience cost as well as performance improvements. Thus if cost is reduced to half and performance only doubled, then we have line C. Now the break even cost point is only 2 x drum up from the 1.5 x drum of line A.

If during that same period, drum cost and performance were also to have improved by the same factors, then one would have line D. Thus under this set of assumptions, the IFD would have to be priced at .15¢/bit if its attachment is delayed to that time frame.

If the above analysis is accepted, what should one conclude about electronic storage? I would suggest that there are three alternatives:

1. Aim for a product costing no more than what one believes fixed head rotating files will cost in the same time frame.
2. Look to replace system components other than drums.
3. Develop or hope for new programming support which can really utilize the smaller blocks and more frequent accesses which these technologies permit.

SUMMARY

Certainly the one parameter equation used above cannot comprehend many of the important contributors to systems thruput. Among the more important of these are the effects caused by special correlations between pairs of system components such as channels and drives. Clutch point phenonema are neglected completely, as are attachment costs. Nevertheless I would suggest that the testing of storage proposals by such a simple model can help provide a rough sizing of the applicability of a given technology as well as establishing a base from which alternative assumptions and product plans may be developed.

ACKNOWLEDGEMENTS

I particularly want to acknowledge many useful discussions with Gene Amdahl and Steve Behman in the development of these ideas.

REFERENCE

1. Hardy, G. H., Littlewood, J. E. and Polya, G., Inequalities (Cambridge University Press, New York 1959).

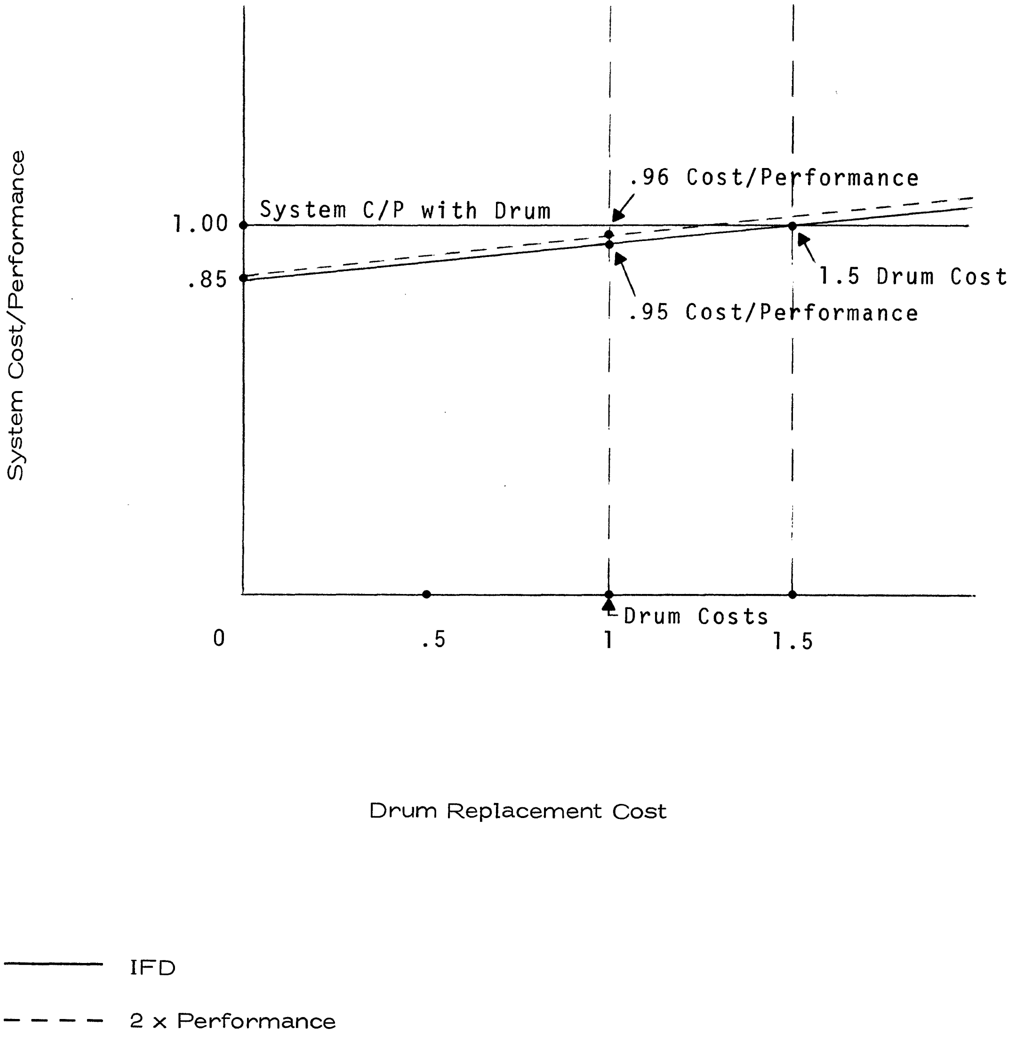
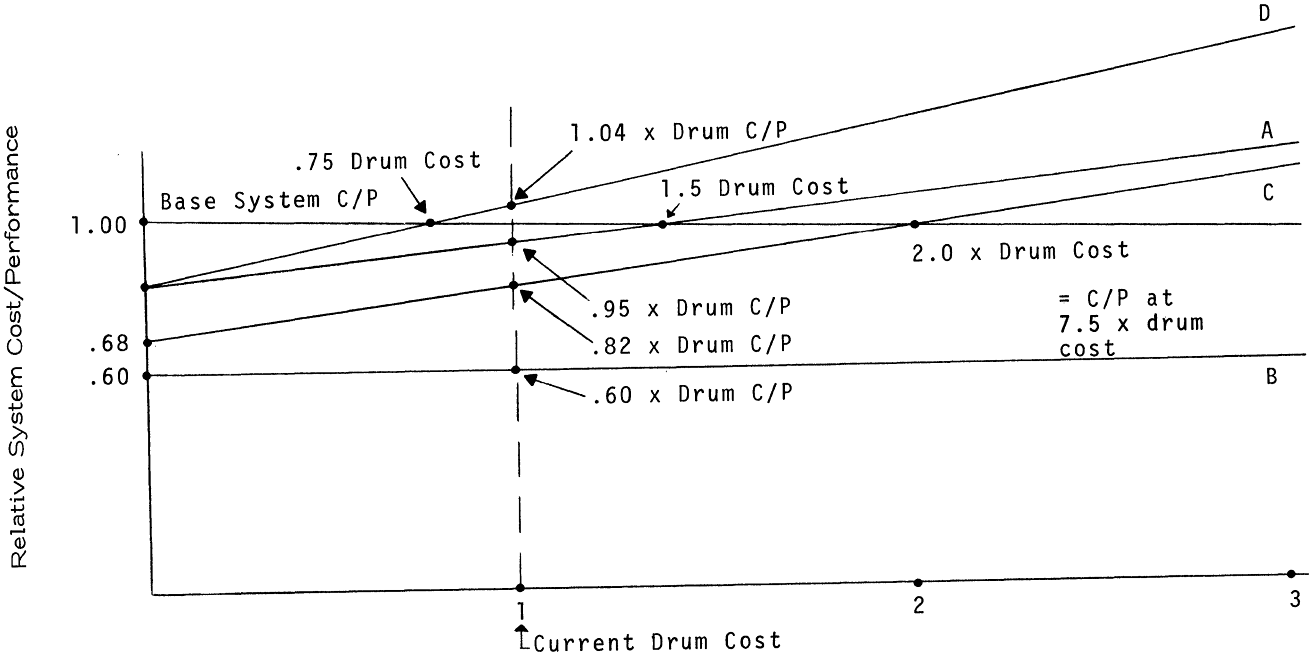

System Cost/Performance
1.00
.85
System C/P with Drum
.96 Cost/Performance
1.5 Drum Cost
.95 Cost/Performance
Drum Costs
0
.5
1
1.5
Drum Replacement Cost
IFD
2 x Performance

FIGURE III



IFD Cost Relative to Drum

A. IFD Replaces Drum in Base System
B. All Other System Components 4 x Performance Improvement
C. All Other System Components 2 x Performance, 1/2 Cost Improvement
D. All System Components (Including Drum), 2 x Performance, 1/2 Cost Improvement

FIGURE IV

DYNABIT[T] MASS MEMORY

B. D. Broadbent, Advanced Technology Department, Hughes Aircraft Company

A new magnetic memory device, the DYNABIT[T] Memory Element, has been designed to fill the gap between core stacks, with rapid random access to a bit and high reliability, and the less expensive techniques that employ a continuous moving storage medium such as discs and drums (see Paper in New Memory Device Session on The DYNABIT[T] Memory). The DYNABIT[T] Memory Element maintains all of the attributes of the continuous storage media memories, but does not have mechanically moving parts and adds a versatility and reliability heretofore available only in linear select systems. The operation of the memory is pictured as analogous to a static drum surface on which the "bits" move instead of the drum. The simplicity of the basic element structure provides some significant new capabilities that greatly simplify many existing data storage problems. The DYNABIT[T] memory can cover a wide variety of functions with costs that vary from five cents-per-bit for special storage and buffering applications to .05 cents-per-bit for mass memory applications.

One of the basic problems in maintaining a high data processing rate when a serial access memory is used is that of avoiding or overcoming the inherent latency or delay in accessing the desired physical record. This problem is of a very different nature with the DYNABIT[T] memory than with rotating memories. There are a number of generally accepted techniques for reducing disc and drum latency that are effective. However, these techniques are expensive in terms of processing capability and in the additional buffering which they require within the computer. Like the disc and drum memories, the DYNABIT[T] memory has an inherent access delay. Unlike them however, the problems associated with delays are much fewer and the penalties which are paid to reduce the effects of the delays are much less. This is largely due to the fact that the DYNABIT[T] is inherently a shift register. Because of this, it is self-buffering and thus much of the core buffering required for reducing the effects of access delays in discs and drums is not needed when the DYNABIT[T] is used. In addition, and just as important, the delays are small and depend mostly on the size of the data block or segment required for the particular system application being considered. For example, the access time or delay necessary to select any block or segment within an entire DYNABIT[T] memory, regardless of the size of that memory, is 20 µsec to the first word. After the block has been accessed the word rate within that block is variable from 0 to ± 250,000 words per second. The plus and minus notation is used here to indicate that the DYNABIT[T] information can be moved forward or backwards on command. The "word" transfer rates may also be stopped and held at any particular position and then started again instantaneously to bring out the next word in that block or segment with only a 1 µsec delay. A typical 2048 block or segment organization utilizing the DYNABIT[T] would have an average word access time of 4 milliseconds. For this organization, this would become a fixed access time and would now be independent of the size of the memory. This allows a growth factor that is not readily available in discs and drums. A particular size disc or drum cannot increase its storage capacity too easily without redesign and this will usually affect the access and latency times.

The DYNABIT[T] memory can be fabricated with a wide range of characteristics and the elements can in turn be interconnected to produce almost any desired organization. If larger blocks or segments are desired, the average access time would increase by 4 µsec for each additional word or set of words. Data transfer rates can be essentially as high as desired, even though the maximum bit rate of a single band is limited as a function of the wire characteristics. Because it can function asynchronously, it can work readily into conventional serial, logical, and arithmetic configurations. Random access to the block level, data pre-ordering for instant access, look-ahead without tying up memory controls, and reverse data flow all enable significant reductions in access time and number of accesses required. The capability to accept and store data at one rate, and discharge it at another, leads to applications such as buffering data exchange between multiple conductors. The DYNABIT[T] memory is an extremely flexible, highly reliable, low maintenance, non-mechanical storage device, available for a wide variety of memory applications. Its economic feasibility has been demonstrated. It is presently in production and should join drums and discs as inexpensive, compact, bulk storage devices of moderate speed and also fill a number of I/O buffering and display applications in a manner that will reduce overall system complexity with improved reliability. In its present state of production memory systems of from 10 to 20 million bits are available at a cost of approximately .5 cents-per-bit. Over the next two years it is anticipated that this cost will be reduced to the area of less than .1 cents-per-bit. Extensive data reliability information is not yet

---

[T] Hughes Aircraft Company Trademark

available for this new technique, however data losses of less than one error in 10^{12} characters for any block have been demonstrated. There is virtually no maintenance involved in the operation of a DYNABIT^{T} memory since it is a solid-state device which is relatively impervious to the elements. High reliability is assured with no moving parts and temperature coefficients of all parts closely matched.

SEMICONDUCTOR I/O STORAGE

R. H. F. Lloyd, Advanced Memory Systems, Inc.

Advanced Memory Systems recently announced SSU (Semiconductor Storage Unit) is the first all-semiconductor memory designed and priced for that price/performance gap which presently exists between all-electronic storage, i.e., cores and electromechanical storage - drums and discs.

The SSU is built out of the lowest cost form of semiconductor memory. That is, metal oxide semiconductor - MOS - shift registers. The low cost results from the fact that this form of memory has minimum electronic support requirements in that decoding is done not to the bit but to a data loop, which contains 256 bits of storage. This design approach also results in minimum packaging costs because the number of I/O pins per bit of storage is significantly lower than a randomly addressable memory. The compromise to achieve these low costs is performance. The average access time of 130 microseconds is slow in comparison to present-day, all-electronic memories but very fast when compared to the 2.5 ms of high speed electromechanical storage. Since this technology costs considerably less than core to manufacture, it is a natural fit between these two technologies and therefore will replace a large number of the present applications for high speed drum.

There is every reason to expect that the previously experienced rapid reduction of manufacturing costs for integrated circuits and integrated circuit memory, in particular, will continue in the future. This rate is significantly greater than that that has been experienced on both core memories and electromechanical storage. It is therefore the conclusion that the SSU approach to high speed I/O storage will become even more favorable from a cost/performance viewpoint in the future, and it will represent the dominant technology for this class of storage applications.

The SSU is built up from a large number of 256-bit data loops or re-circulated shift registers. The internal organization is such that each storage card contains one bit position for all the words in that section of the memory, i.e., bit organization. This allows full advantage to be taken of error correction such that any component on a card or the total card can fail and still be corrected by single error correction. The SSU is built with single error correction and double error detection. Further, since the power dissipated in a shift register is directly proportional to the shift rate, this approach allows significant reduction in total system power. This is achieved by normally re-circulating all the data loops in the SSU at a very low shift rate. When addressed, only the data loop containing the desired information is brought up to a very high speed shift rate thereby facilitating a short access time and high data rate. Since this represents a very small percentage of the total storage and occurs on relatively short duty cycles, the total average power of the system is determined primarily by the low shift rate not the high shift rate. Two internal counters are used so that synchronization can be maintained after the addressed loop is reduced to the lower shift rate. This approach results in an average power of 500 watts for a 2 million byte memory and therefore very low operating temperatures. Thus, significantly enhancing the reliability of the memory. A mean time between failures of 2,000 hours for a two megabyte unit is projected using this approach. Further, 94% of the hardware can be protected by error correction. A very conservative estimate of total system reliability with ECC is 5,000 hours MTBF. Eight million bytes of storage can be housed in a frame 30" by 60" by 70". This is approximately 8.3 times the density achievable with core and 1.4 times the typical high speed drum system. The end-user price for the SSU is 1.98¢/bit for a two megabyte unit and 1.0¢/bit for a 16 megabyte unit. It competes today with high speed drums from a cost/performance standpoint and therefore will displace drums where access time is critical. The cost/performance comparison is particularly favorable when record gaps are considered. For records in the 500 byte area, as much as 45% of a high speed drum's capacity is consumed in overhead. Since this overhead is primarily record gap, the lack of the record gap requirement in the SSU almost halves the effective cost/bit for these very typical record sizes. For the next two or three years, drums or fixed head discs will remain slightly cheaper and a lot slower. If significant improvements are made in track and bit densities, this situation would exist indefinitely, thereby guaranteeing drums a permanent but smaller position in the hierarchy of storage devices.

A BRIEF REVIEW OF CURRENT DESIGN TRENDS & ASPIRATIONS IN ELECTROMECHANICAL MASS STORAGE

R. Tickell, Senior Member, IEEE
Univac Division of Sperry Rand Corporation

The design specification of an ideal, practically realizable mass storage file would probably be built around a fixed-head drum or disc were it not for the high cost of recording heads involved. Contemporary non-contact recording head design is based upon a recording surface speed of up to 2500 in/sec. This figure corresponds, for example, to a 20-inch diameter disc driven at 2400 rpm so as to provide a 12½ ms average (25-ms maximum) access time for all data stored on that disc.

All of this is not to say that 12½ ms is the answer to a designer's prayer; it merely emphasizes that, given the low cost head, very respectable access times could be achieved in contemporary design. Of course, future improvements can also be expected in other areas that have a bearing on this aspect of performance.

A real problem in the continuing search for an economically-priced head concerns reduction of physical size that will permit track densities to increase substantially. Otherwise volumetric efficiency of the overall storage file will be far short of the potential that is offered by present day oxides and plated media. We have grown accustomed to thinking of pulse densities greater than 2000/inch along the length of a track. But there is nothing in the composition of the recording medium to dictate that track density must be held to a figure that is one order of magnitude less.

So, pending the realization of these design goals in multi-channel, fixed heads, the popular compromise being pursued today centers on single-channel heads working on 14-inch diameter discs and with great emphasis placed on the track-to-track moving time - typically offered at average values ranging between 25 and 50 ms. (Average rotary latency of 12½ ms, corresponding to 2400 rpm, must be added to this figure to calculate a total for "look-up" performance.)

What are the prospects of resolving the head economics problem or the accessor performance problem, or will there be some ultimate compromise? It is most likely that accessor performance is soon to reach a point of diminishing returns at or around the "25-ms average" level. The reason lies in the high accelerations demanded by such performance and the stability of the aerodynamics related to the recording heads. The gap between the access time performance of a fixed head and that of a mobile head is not therefore likely to be reduced to a negligeable amount.

Newer, more innovative avenues of endeavor will be explored in the mobile-head machines in attempts to reduce the overall access time penalty. For example, a more elaborate record search facility will be designed into the device so as to free the control unit from search delays. Storage capacities can be substantially increased through recourse to track-seeking, position-sensing mechanisms. Certainly, the accumulation of tolerances that go with interchangeable disc packs seems to make this line of development a necessity.

Both fixed and moving-head files can benefit from new encoding techniques aimed at the maximization of data-packing density without commensurate increases in density of flux reversals. It is unnecessarily restrictive to have to decide individual flux reversals rather than pairs, triples or even bytes. By way of illustration, one may borrow an example from the communications world where a 100% understanding of a voice transmission is not necessarily dependent upon a 100% complete reception of each individual consonant and vowel - or even syllable. The listener himself clarifies distortion through previous knowledge of language rules and patterns. The next few years will surely bring developments along these lines, aimed at the accommodation of lower signal fidelity due to increased recording density.

There is no doubt that the use of so-called contact recording heads is a great aid to data recovery techniques. It does, unfortunately, require extra special care in the surface finish of the recording medium as well as the avoidance of excessive run-out so that head/disc contact may be preserved or, in other words, flying discouraged. This is seen to be a more serious problem in the larger assemblies, for example, in discs or drums exceeding 24-inch diameter and/or 2500 inch-per-second surface speed.

The cost of maintenance is a serious burden that grows to a higher level in electromechanical devices than it does in purely static devices. Fault diagnosis and repair of a high performance servomechanism is a formidable undertaking when both mechanical and electrical components are considered, and a high degree of specialization is demanded of the service engineer.

To summarize, there is still much room for further development of random-access mass storage as we now know it. The needs of extremely large capacity users, for example, insurance companies and record-keeping governmental agencies, will continue to grow to monumental proportions. At the same time, the cost of such storage will be expected to decline in keeping with the well-established idea of economy of scale. Currently, the cost to the end user of storing one data bit in a high-performance, mobile-head machine is around 0.01¢. The cost of head-per-track storage, on the other hand, ranges from 0.05¢ to 0.10¢ depending on the level of performance and capacity demanded. (The price range reflects the better volumetric efficiency of slower speed units operating at 1200 rpm or thereabouts, while the more expensive machines may be operating at 2400 rpm and up with corresponding reduction in access times.)

Large users will expect to see at least one order of magnitude reduction in price of storage during the next five years or so. That such performance might be achievable by any other means than those outlined above is difficult to imagine, and the prospect of a substantial increase in complexity of mechanism in order to achieve the desired levels of performance is not a happy one for the field service engineer.

HEAD PER TRACK DISK FILE MASS MEMORIES

R. W. Jack, Westlake Plant, Burroughs Corporation

The most significant characteristics of mass storage devices are cost per bit of information stored, reliability, average access time, and information transfer rate. This paper describes the trend of some of these characteristics over the next five years. It is concerned with fixed head disk storage devices of the head per track type.

Cost Per Bit

Figure 1 presents a forecast of cost per bit (¢ per bit) for head per track disk file sub-systems for the years 1970 thru 1975 and for average access times of from 40 down to 5 milliseconds. Average access time is defined to be one-half the average rotational period of the disk. The discussion will develop the point that the continual decrease of cost per bit is due to increased bit packing densitites, increased track density, and lower manufacturing costs.

Reliability

For the purpose of this discussion, reliability is described in terms of the recoverable error rate and the non-recoverable error rate that one would expect from a typical product of this type over a long period of time. Figure 2 describes the error rates for disk pack storage devices and head per track storage devices. The disk pack figures are those being quoted by some manufacturers at this time for the 2311 type disk pack drive. The error rate for a 4400 bit per inch/200 track per inch disk pack storage device (which does not presently exist in the market place) is estimated. This estimate is that of the author. The 1970 error rates for head per track storage is that being achieved today. A forecast is made for this type of disk storage thru 1975.

Transfer Rate

For this discussion transfer rate is defined as the byte (8 bit) transfer rate at the input/output unit. At this time transfer rate limits appear to be with the central processors and/or I/O channels. Head per track disk storage devices today have byte transfer rates of approximately $.4 \times 10^6$ bytes per second. In one instance, a file which will be delivered this year (the ILLIAC Disk File Sub-System) will have an effective byte transfer rate of over 60×10^6 bytes per second. At this point in time, a better definition of system need for these high transfer rate files is needed.

Summary

We have just started really exploiting the full capabilities of head per track disk files. These files will remain competitive over the next five years in all aspects of cost and performance characteristics. The simplicity of this file with respect to software and input/output control design is an important advantage. Further the reliability of this type of file should always have an order of magnitude edge on removable medium storage devices.

HEAD/TRACK FIXED HEAD STORAGE
COST/BIT (¢ BIT)
STATE OF ART FORECAST

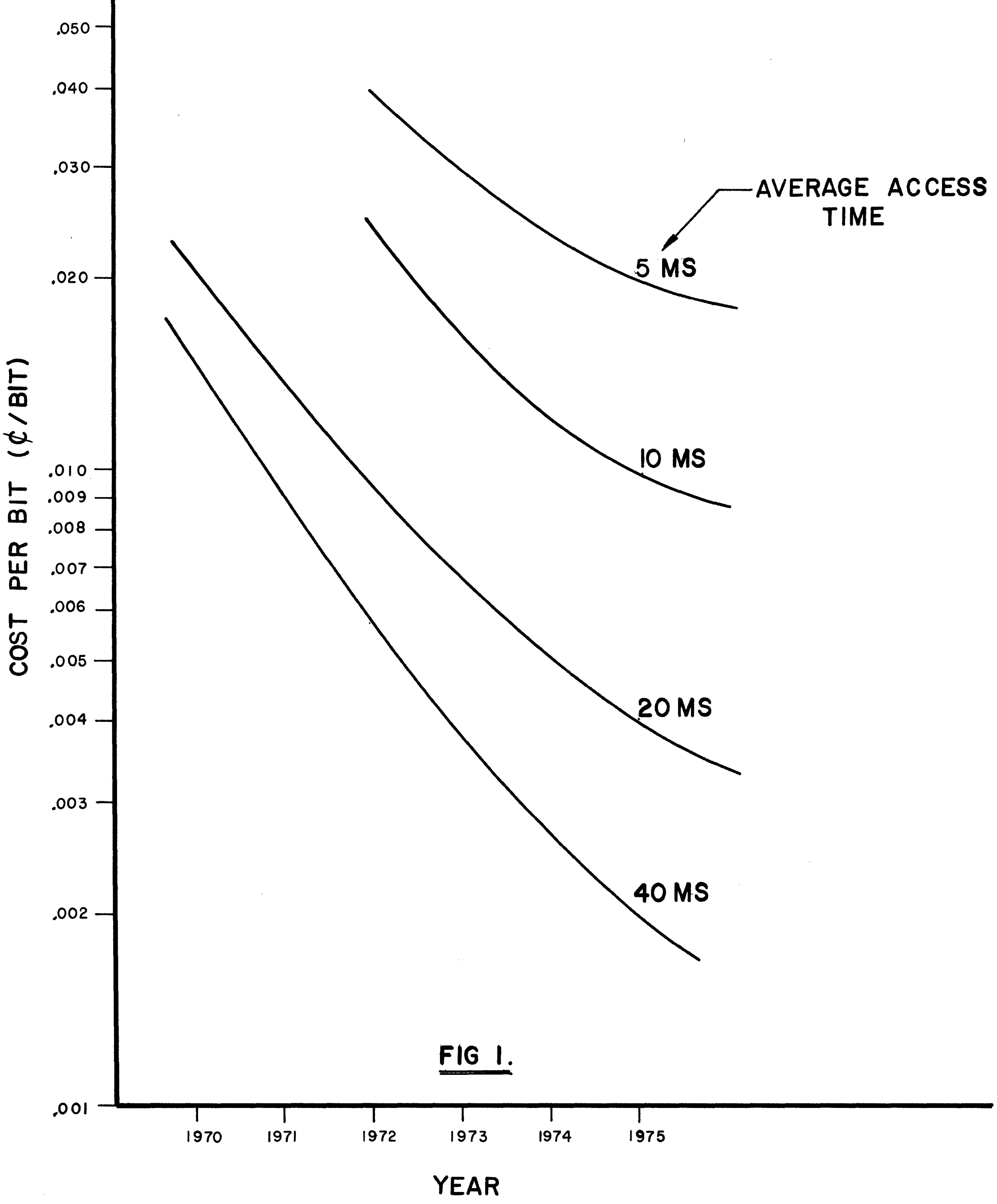


FIG 1.

Reliability

	Recoverable Error Rate	Non-Recoverable Error Rate
Disk Pack Storage 2311 Type	1 in 10^{11} bits	1 in 10^{12} bits (not attributed to disk)
(2200 bpi/100 tpi)		
(4400 bpi/200 tpi)	1 in 10^{10} bits	1 in 10^{11} bits (not attributed to disk)
Head/Track Storage 1970	1 in 10^{12}	1 in 10^{14} (For any reason)
1972	1 in 10^{11}	1 in 10^{14} (For any reason)
1975	1 in 10^{10}	1 in 10^{13} (For any reason)

Figure 2

MOVEABLE HEAD, FIXED DISC SYSTEMS

Irving L. Wieselman, Data Products Corporation

INTRODUCTION

The cost and performance characteristics of rotating media mass memories are dependent on the functional design characteristics of the system. Various approaches to the design problem have resulted in memories with either fixed or moveable heads with removeable or fixed media. It is the position of this author that, for the class of memory with access times in the 10 to 100 millisecond region, the moveable head, fixed disc media system offers the greatest reliability, lowest cost per bit and lowest total cost.

COST AND PERFORMANCE RELATIONSHIPS

The primary factors for the choice of a storage device for a computer system are access time, transfer rate, reliability, cost/bit and total cost. Storage devices, in general, possess the property that the lower the access time, the higher the cost/bit. For large capacity storage, the system designer chooses the device with the lowest total cost consistent with necessary performance.

Rotating storage devices in the 10 to 100 millisecond access time region have unique performance properties which would cause them to be selected for use. Minimum access time is achievable with the fixed head disc system, since no time is required for moving heads to the desired track. The minimum off-line cost per bit is achievable with the moveable head, removeable disc system, since the cost of a disc pack is low. The minimum on-line cost per bit is achievable with the moveable head, fixed disc system.

The effective access time for a moveable head system can be the same as a fixed head system when overlap seeks are used in a multiple positioner system. At higher cost, the performance advantages of the removeable disc over the fixed disc relate to the off-line replaceable storage capability and lower latency times.

FACTORS AFFECTING COST

The storage capacity of a disc is the product of the number of bits per track and the number of tracks. Moveable head systems use at least two orders of magnitude fewer heads than fixed head systems, since the positioner moves the heads to tracks rather than requiring one head per track. For an equivalent number of tracks, a positioner and its heads are less costly than an equivalent number of heads. Published prices indicate that a fixed head system with one-third the access time of a moveable head system costs almost three times as much per bit.

Storage is less costly in a fixed disc system than a removeable disc system. A fixed disc system can use larger discs (24") than a removeable disc system, since the accepted standard for removeability is 14". Additionally, the tolerances for the removeable discs are more stringent than for fixed discs. Hence, for positioning devices with comparable tolerances, higher track densities are achieved with the fixed disc system. The tracks can also store more information, since larger circumferences are involved. Since larger discs store more information, fewer heads, discs and positioners are required than with the smaller removeable disc system. Therefore, the cost is bound to be lower. (See Table I.)

A standard reference should be used when comparing costs, such as the users price (shown in Table I) or the manufactured cost. The systems manufacturer buys in large quantities from the peripheral manufacturer for a price of 50% to 70% of the users price. The peripheral manufacturer would like to sell at twice the manufactured cost (labor, material and manufacturing overhead). Thus, the manufactured cost is 25% to 50% of the users price. The cost of engineering, software development, marketing and administration must be added to the manufactured cost. This leaves a typical profit of between 10% and 20% before taxes.

FACTORS AFFECTING RELIABILITY

Storage reliability is measured by the mean time between component failures, the mean time to repair and the ability of the system to read and write information reliably. When considering systems with components of comparable reliability, the system with fewer components will have the lower MTBF. Another factor in disc file reliability is the cleanliness of the file and the ability of keeping humans away from critical areas. Minimum tolerance changes in components provide maximum read/write reliability.

The most reliable system should be the servo-positioner driven, moveable head, fixed disc system, since: (a) moveable head systems have two orders of magnitude fewer heads than fixed head systems; (b) moveable head, fixed disc systems have less than one-half the components of a removeable disc system; (c) fixed disc systems, unlike removeable disc systems, do not require humans to get near the disc head interface and are not subject to the tolerance changes in moving packs from spindle to spindle; (d) the maintenance of constant temperature possible in a fixed disc system should minimize positioning and head tolerance changes; (e) a servo-positioner does not wear like a positioner with detents.

Based on existing field operating equipment and on technological advances, the expected reliability of the moveable head, fixed disc system is an estimated MTBF of between 2000 and 5000 hours. The MTTR should be between one-half and one hour. Non-recoverable or hard errors can only occur because of a component failure, but not all component failures cause hard errors. The recoverable read error rate should be between 1 in 10^{9} to 1 in 10^{10} bits passed. Write errors recoverable by rewriting have an error rate of 1 in 10^{11} to 1 in 10^{12} bits passed.

PRICE PER BIT AND TECHNOLOGY

The users price per bit for moveable head rotating memories today is between .005¢ and .015¢ per bit. Future systems will reduce the price per bit by maximizing bit and track density through the use of more effective components, not by increasing their number. The effect of technology on price will be discussed relative to moveable head systems.

The limiting factor in track density is the minimum width track which can be read. This factor is a function of the magnetic properties of the disc, flying height of the head and manufacturing tolerances. On a moveable head system, the next most important factor is the accuracy of the position of the head relative to the track. The track spacing is equal to the minimum width read track, plus 3 to 5 times the total head positioning tolerance.

Today's moveable head systems utilize minimum read width tracks of 3.5 to 5.0 x 10^{-3} inches. Head positioning accuracies are in the 0.5 to 1.5 x 10^{-3} inches. Track densities vary from 100 to 150 tracks per inch. Practical systems within the next five years will be capable of minimum read track width of 2 to 3 x 10^{-3} inches. Positioning accuracies are capable of an order of magnitude increase by means of servo-positioners which utilize information on the discs as position reference information. Track densities should then range from 200 to 400 tracks per inch.

Bit densities in today's moveable head systems range from 1500 to 3000 bits per inch. Higher bit densities are achieved by flying the heads lower and by using more uniform magnetic materials. A factor of 2 increase in bit densities should be available during the next five years. More effective coding schemes than simple phase modulation might also increase the effective bit densities by another factor of 2.

Thus, the increase in packing density should go up by a factor of 4 to 10 in the next five years with the same reliability as today. Since these densities will be achieved by improvement in component technology, the cost for components should go up by no more than 25%. This means that the cost per bit should decrease by a factor of 3 to 8 in moveable head systems.

TABLE I

	Data Products 7314	IBM 2314
Type	Fixed Disc	Removeable Disc
Compatible Capacity	233.4×10^6 bytes	233.4×10^6 bytes
Maximum Capacity	274×10^6 bytes	233.4×10^6 bytes
Cylinder/Disc Module	800	200
Disc Diameter	24"	14"
Tracks/Inch	150	100
Bits/Inch	1800	2200
No. of Discs	20	80
No. of Heads	80	160
No. of Positioners	2	8
Type of Positioner	Servo	Detent
R.P.M.	1800	2400
Avg. Rotational Delay	16.7 ms.	12.5 ms.
Transfer Rate	234,000 bytes/sec.	312,000 bytes/sec.
Avg. Seek Time	60	60
Approx. Price	$98,000	$163,000
Approx. Price/Bit (Max. Capacity)	.0045¢	.0087¢

DESIGN DIRECTIONS FOR ROTATING MASS STORAGE

Harold S. Yang, Information Storage Systems

ABSTRACT

Removable-Media Disk Files have become increasingly popular as direct access storage devices. This paper summarizes some of the basic performance characteristics of present and future Removable-Media Disk Files and discusses some of the design considerations involved in improving the associated magnetic recording technology.

ROTATING MASS STORAGE

The basic performance characteristics of any rotating mass storage device are access time, cost per bit, capacity, "latency" (rotation speed) and data transfer rate. IBM data compatibility requirements will directly affect the last three characteristics.

Each of the three classes of rotating mass memories has its particular class characteristics. At the risk of generalization these are:

1) Fixed-Head Files (Drums and Disks)
    a) High data transfer rate
    b) Short access times
    c) High cost per bit
    d) Small capacity

2) Movable-Head Files
    a) Large capacity
    b) Longer access times
    c) Low cost per bit

3) Removable-Media Files
    a) Medium capacity (expandable)
    b) Medium access times
    c) Moderate cost per bit

Of the three classes of rotating mass memories, the Removable-Disk Memory has received the most attention of late. This popularity is due primarily to the fact that the file medium is removable. This fact allows the user unlimited off-line storage (like tape) and minimizes the requirement for "backing up" the file data to continue system operation if a drive breaks down. That is, if a drive does malfunction, the operator switches the disk pack from the failing drive to another working drive and continues system operation, possibly at a reduced activity level.

Approximate characteristic values for Removable-Media Files are as follows:

Capacity - 250 x 10^6 to 1 x 10^7 bits

Access time (average) - 42.5 milliseconds (30 millisecond positioning time plus 12.5 millisecond rotational latency)

Cost per bit - .005 cents per bit.

In addition, field experience has indicated that the following reliability performance can be expected of a well-designed and manufactured Removable-Media Disk Drive. Typically, around three or four calls per year by a trained serviceman to repair an operational malfunction are required. Typical repair times for a competent serviceman are under two hours.

Error rates are very difficult to evaluate because many factors contribute to errors in a Removable-Media File. However, excluding errors attributable to the disk pack (such as defects, scratches, etc.), and operating in a non-hostile environment, recoverable error rates of less than one bit in 10^9 are common.

DESIGN DIRECTIONS

The overriding design consideration involved in improving magnetic recording technology is cost. Increases in bit density come at the expense of more precise tolerances on all recording components. Consider, for example, the relationship between flying head altitude and bit density.

		NOM INSIDE
2311	(1100 bpi)	120 µinches
2314	(2200 bpi)	80 µinches
XXXX	(4400 bpi)	50 µinches

Clearly any increase in the bit density requires an increase in head component precision.

In addition the recording media must undergo certain changes. The disks must provide a structurally more stable flying platform for the heads. The smaller recording gap of the higher bit density head dictates the geometry with respect to flying height and media thickness. Since the leakage field of the head must magnetically saturate the surface, a smaller gap dictates a lower flying altitude and a thinner magnetic coating. The lower signal levels associated with higher bit densities may cause drive designers to package IC amplifiers out on or near the head itself. Actual circuit design will probably not change much. Instead components will be located specifically to improve signal to noise ratios. Any attempt to increase the "resolution" of the recording technology must place primary emphasis on the recording elements themselves. Consequently, the largest expense in improving the technology is always the cost of developing the heads and the media.

Many coding schemes for magnetic recording exist today. The most commonly encountered method in removable media files is the NRZ Double Frequency method. Double Frequency is the term given to the recording system that injects a clock bit at the beginning of each bit-cell time, thereby doubling the frequency of recorded bits. The principal advantage of Double Frequency recording is that it is self-clocking. In addition, no AGC is required as in NRZI recording. Consider the three coding schemes depicted in Figure 1.

Line 1 represents the recorded transitions for Double Frequency (2f) recording. The transition density is twice the information density due to the clock bit injection requirement. Line 2 represents NRZI recording. Here a transition occurs only for a "one" (data bit present). Line 3 represents Modified Frequency Modulation recording (MFM). Here a clock bit is injected if and only if there are "no data bits on either side." Since clock bits are recorded, MFM has the advantage of being self-clocking. Note, however, that unlike Double Frequency recording, the transition density for MFM is the same as the information density. Consequently, it would appear that using MFM encoding we could double the recorded information density over 2f recording for a given level of Head-Disk technology.

There are problems involved with this approach. MFM, like 2f, requires that the data bits be separated from the clock bits. Note that for 2f, the "data window" is 200 nanoseconds. In other words we must "look for a clock bit" every 200 nanoseconds. Since clock bits are recorded in MFM, for the case shown the data window is also 200 nanseconds. If we attempt to increase the recorded information density by "doubling the frequency," we halve the size of the data window or, in other words, cut our "clocking tolerance" in half. To see this more clearly, observe that if we double the frequency, we must differentiate between 200, 300 and 400 nanosecond gaps instead of 400, 600 and 800 nanosecond gaps. Generally speaking, for a given technology, attempts at information packing make recovery of the recorded data more difficult.

Increased bit densities require that defect size on the media be reduced. If this does not occur error rates, both hard and soft, will rise. This increase in error rates may force

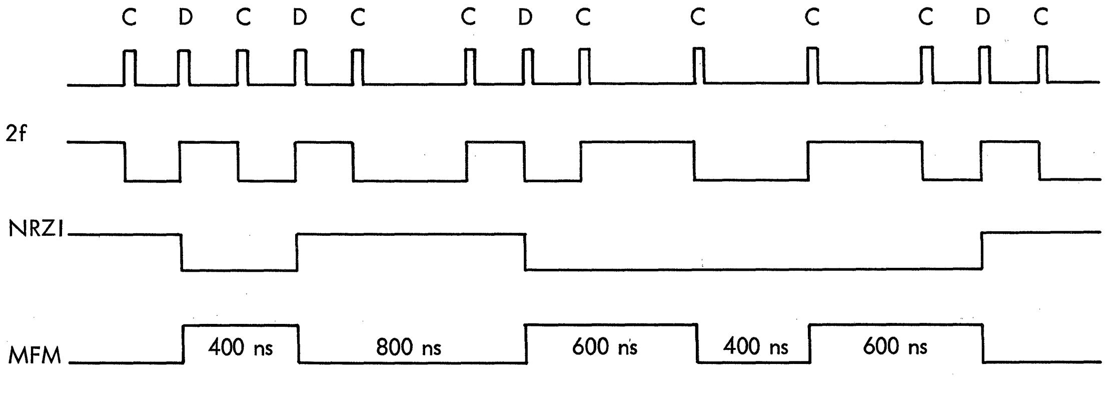


Figure 1

the adoption of error-correcting codes to allow us to live with the defects. Error correction, however, cannot be used to compensate for an inferior technology. Its primary value lies in its ability to compensate for anomalies in the media.

Track density on interchangeable media disk drives will probably be improved by utilizing track-following techniques (positioning on recorded signal). Doubling of present track densities (100 tpi) certainly will occur within the next few years with 500 tpi not too far down the road.

Voice Coil Actuators and closed loop servo control have more than halved the access time of removal media disk drives in the past year (ISS 701, 714). It is expected that the present access times will be halved again (from 30 ms to 15 ms) within the next few years. As the access times of the removable media drives become short with respect to rotational time they will start to replace drums and fixed head files in many applications.

In summary, removable media drives appear to be the glamour member of the family of rotating mass memories. Improvements in the recording technology and positioning speed and accuracy come at the expense of more precise tolerances on all componentry involved.

PARALLELISM IN SYSTEMS ARCHITECTURE

D. L. Slotnick, Department of Computer Science, ILLIAC IV Project
University of Illinois

Introduction

The two main streams of thought in the mind of the computer designer when he thinks about computer architecture concern the problems he wants to solve and what the technology gives him to work with. Clearly, there are no absolutes here. First, he is interested for one reason or another in only a subset of the world's problems. Secondly, he is only aware of a subset of the relevant computer technology and moreover applies his individual experience and judgment to evaluating this technology. My own problem interests and my own awareness and evaluation of technology is what I will discuss here. The only claim that I make for the discussion below is that it logically relates the problems in which I have interest and the technology of which I am aware with the architectural approach I have selected to pursue.

1. Computer-based Considerations

Small signal switches have come down in price over the past several years by one to two decimal orders of magnitude. Simultaneously, their speed has increased by at least a comparable amount and likewise their reliability. In memories, peripherals and programming, the situation has remained relatively stable. The inescapable conclusion is that the design may employ switches rather lavishly if it yields significant improvement in solving problems. There is considerable disagreement with regard to how cheap cheap is, how fast fast is, and how available available is. Fortunately, the conclusions I wish to make do not require me to add my detailed opinions on these questions.

Semiconductor memories are here. As opposed to magnetic memories, be they core, film, wire or what have you, these memories have a cost which is relatively size-independent. This opens the door to high bandwidth, multiport exchange with logic and peripherals. Semiconductor memories, moreover are fast. The conclusion is that only very fast processing rates can take advantage of these characteristics. Semiconductor memories aren't cheap which implies the necessity for continuing to design systems with hierarchies of storage.

While there has been a thoroughly enjoyable increase in the bit densities of magnetic secondary storage units, the main improvements have come from employing additional linear circuits and logic circuits with these units to make them more effective in the system. The availability of a new class of tertiary storage of impressive capacities (10^{12} bits) with access times two orders of magnitudesless than the time required to go into the tape vault, find a tape, wash your hands and hang it, and with transfer rates that are reasonable (several megahertz) is an event which a system designer must also take note of.

Other electro-mechanical system elements have gone roughly the same route. There have been nice improvements in the effectiveness of system employment of these devices accruing primarily from judicious use of additional associated channel electronics.

2. Problem-based Considerations

It has been observed by many that the society in which we live fails of absolute and final perfection. The use of natural resources and municipal and regional planning are instances, I believe, where the application of man's unaugmented endowment of intelligence has produced crisis situations. It is my contention that computer technology can play an important and perhaps decisive role in supplying to those who plan on a large scale precisely those capabilities their genetic endowment does not.

Many problems in resource allocation are currently being solved by linear programming. With sufficient computing means this technique can be applied on an economy-wide basis. An effort currently underway is attempting to do this for the entire agricultural-economic sector of a society. With sufficient flexibility in the model's formulation to permit it to accommodate the features unique to a developing or developed economy. This problem involves thousands of constraints and on the order of tens of thousands of variables. The requirement to employ agricultural resources in a more optimal fashion is for many nations of the world fully established.

Numerical weather prediction techniques currently produce reasonable quality in a twenty-four hour forecast. If this same quality were available in five day forecasts it would produce a multi-billion dollar benefit to this nation alone. One of the obstacles in achieving this objective is computational capacity. This area can absorb almost arbitrarily great computer capacity increases primarily in processor speed.

Key decision processes in questions such as highway location, urban growth and attendant zoning and transportation and terminal facilities development are performed largely as disconnected sets of ad hoc suboptimizations. With sufficient storage and processing capabilities, one can now contemplate the creation and employment of regional, perhaps even national resource inventories which would permit planning in these areas to be done on a basis which makes use of soil data, geological data, hydrological data, climatological data, demographic data and economic and social values and costs.

Interactions between species in an ecological system has been the subject of intensive experimentation; unfortunately without the benefit of prior simulation. With sufficient scientific effort and sufficient computational capacity, there is no reason why useful numbers and perhaps even scientific insight could not be generated by the simulation of quite complex ecosystems. Work is already underway in detailed modeling of ecosystems containing a small number of species and population dynamic models have been built of considerably more complex ones.

An increasing percentage of the data employed by scientists and engineers is generated automatically by sophisticated instrumentation systems. Array radar, sonar and seismometers are cases in point. Such systems serve interests from national defense to geophysical exploration. They have the common features of requiring the application of exotic filtering algorithms and Fourier inversion to be rendered useful. Such systems have over the years played an increasingly more central role in the accumulation of needed knowledge about the planet and the universe. These systems consume a considerable percentage of our total data processing resources and this percentage will probably not decrease.

3. Parallel Systems

The problems mentioned above have the common feature of requiring the repetitive application of significant blocks of code to many different data. In the case of Partial Differential Equations, mesh points; in the case of matrices, their elements; in the case of inventory search, data records. This, coupled with the technological considerations indicate the computer architecture which is the base of the ILLIAC IV system and which can be extended downward or upward with little intellectual difficulty. I will briefly describe two antipodal points in this architectural spectrum. They have both been described in more detail elsewhere.

At one end of the spectrum is a system composed of bit serial processors each possessing a pair of tracks on a rotating store for its memory. Figure 1 illustrates the logic associated with each pair of tracks on the store. Actually, it intends to illustrate a range of systems where the options to be selected in an application-dependent way are hopefully clear.

Two tracks stream bits to the logic and the resulting bit stream is stored on one of the tracks. The logic is controlled centrally (vertical lines terminating near the top of the page indicate a connection with the central control means). The resulting serial system has very much in common with the processing elements of SOLOMON [1]. Data is read from the track 1 amplifier and transmitted to routing logic. This routing logic can switch the data stream either into the further logic associated with this track or to the corresponding entry points in any of p other (neighboring) pairs of tracks. Following the routing logic output to the box marked "switching and delay," we see that corresponding neighbors are ored together onto this line so that if the data stream from track one is being transmitted to a neighbor, then a corresponding neighbor will be transmitted to the logic unit being considered. Alternately, a data stream may be furnished from a central source. Thus, of the p + 2 data sources ored together into the switching and delay unit, one and only one is active. Track 2 furnishes the second operand stream to the process unit through the switching and delay unit.

The delay amount, in binary, is held in a delay set register which may be local or central. The elementary unit of delay may be a bit time, character time, or word time.

The switching and delay unit switches and delays serial streams by the indicated amount and passes them to a process unit. The process unit possesses a small but logically complete repertoire of arithmetic and Boolean commands, depending on the richness of the set it requires from 30 to 200 gates. The result is transmitted to a track select switch which stores it on either of the two tracks.

A more expensive alternative is to provide each pair of tracks with a small electronic data buffer from a few characters to a few thousand characters. The modifications to accomplish this are shown by the dashed lines. To do so for each of 1000 (roughly) pairs of tracks makes the system expensive but much more powerful.

---

[1] Borck, W. C., McReynolds, R. C., and Slotnick, D. L., "The SOLOMON Computer," Proceedings of the 1962 Fall Joint Computer Conference, pp. 97-107, Spartan Books.

Each processor can be inhibited by a local data-dependent value (called "mode" after SOLOMON) which can inhibit the storage of a result stream for any particular track pair, thus providing means for local control. The mode register may be one or more bits in length. Four is a nice number for the length of the mode register.

The control philosophy of the system must, of course, be to generate all major epochs from a clock track on the disk. It would be chaos to attempt anything different. Considering reasonable disks, bit streams would be provided to the logic at a 1 to 5 MHz rate. This would correspond to a gate propagation requirement of 10-50 ns if one is to accomplish 20 stages of logic. This is a comfortable range for today's gates and a few hundred gates of this speed can be integrated on a chip set in close proximity to the heads which feed it. This closeness produces a reasonable requirement on the amplification that makes the whole proposition feasible.

It has become fashionable to provide means to get information in and out of a system which contains storage and logic. Some fashionable boxes are illustrated in Figure 2. There are two genres of currency in this system: data streams and long words each of which represents a single bit of the totality of all mode registers and thus is on the order of 1000 bits long. Associated with the mode registers is logic which executes a specific repertoire on these long words. Among these are counting the number of ones, detecting a long 0, and detecting one and only one 1. Also illustrated in Figure 2 is a buffer capable of storing several sectors or tracks from the disk for transmission to or receipt from any other associated major electronic units such as a computer.

At the other end of this spectrum is the ILLIAC IV computer with its 64 floating point (12,000 gate) processors each possessing an individually indexable 2,000 word memory of 64 bits words. This system has been fully described in the literature and is in the construction phase.

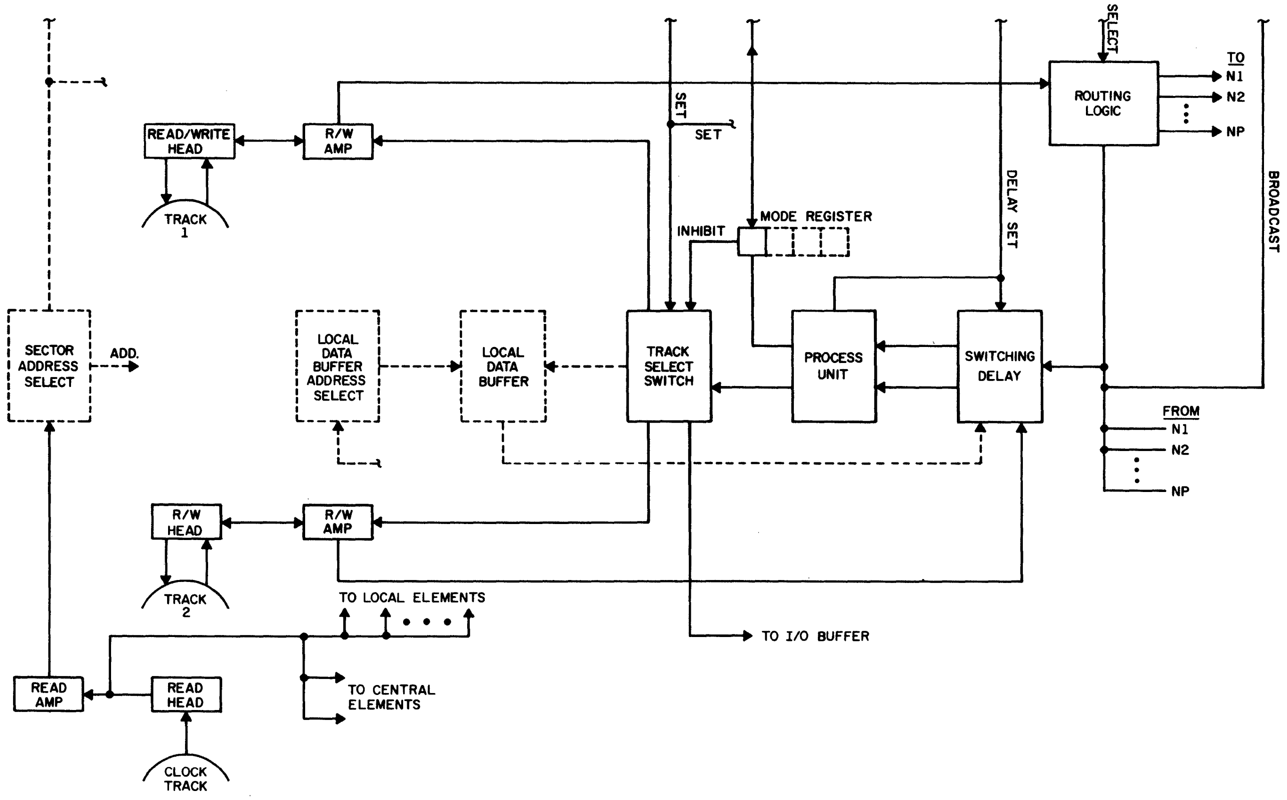

SECTOR ADDRESS SELECT
ADD.
READ/WRITE HEAD
TRACK 1
R/W AMP
SET
SET
INHIBIT
MODE REGISTER
DELAY SET
SELECT
ROUTING LOGIC
TO
N1
N2
NP
BROADCAST
LOCAL DATA BUFFER ADDRESS SELECT
LOCAL DATA BUFFER
TRACK SELECT SWITCH
PROCESS UNIT
SWITCHING DELAY
FROM
N1
N2
NP
R/W HEAD
R/W AMP
TRACK 2
TO LOCAL ELEMENTS
TO I/O BUFFER
READ AMP
READ HEAD
TO CENTRAL ELEMENTS
CLOCK TRACK

Figure 1. Logic per track (pair).

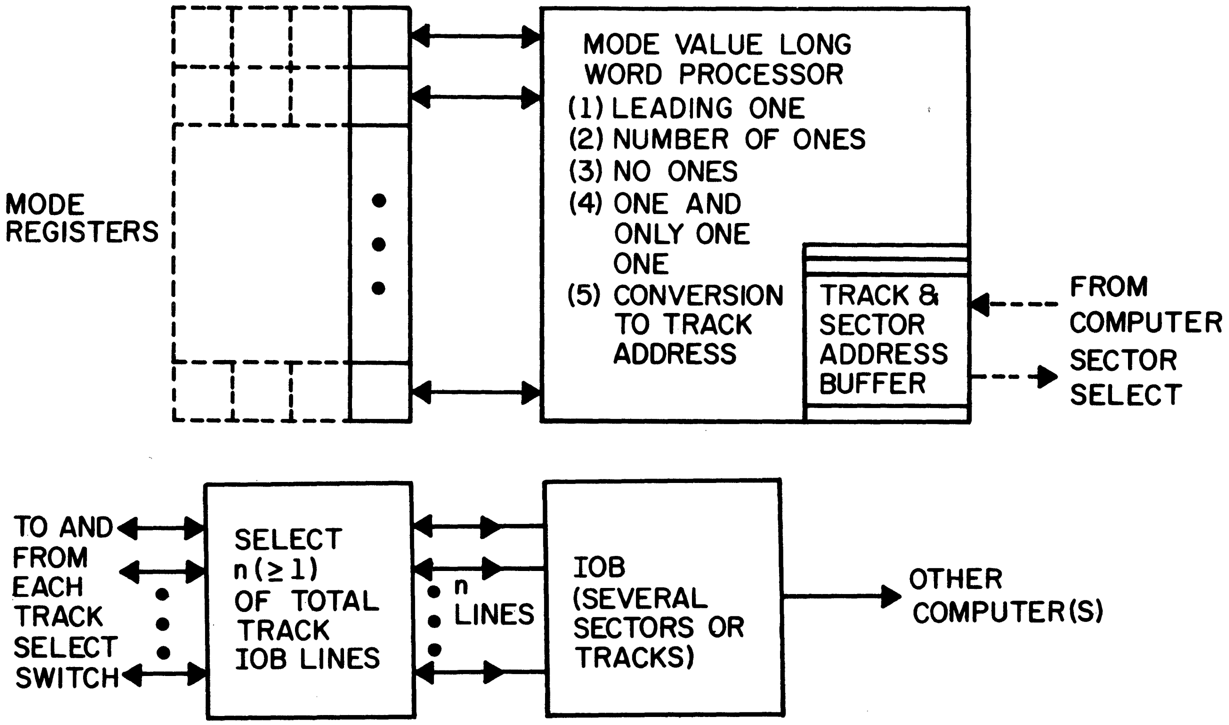

MODE REGISTERS
MODE VALUE LONG WORD PROCESSOR
(1) LEADING ONE
(2) NUMBER OF ONES
(3) NO ONES
(4) ONE AND ONLY ONE ONE
(5) CONVERSION TO TRACK ADDRESS
TRACK & SECTOR ADDRESS BUFFER
FROM COMPUTER
SECTOR SELECT
TO AND FROM EACH TRACK SELECT SWITCH
SELECT n(≥1) OF TOTAL TRACK IOB LINES
n LINES
IOB (SEVERAL SECTORS OR TRACKS)
OTHER COMPUTER(S)

Figure 2. Interfaces.

STORAGE AND I/O PARAMETERS AND SYSTEMS POTENTIAL

Dr. G. M. Amdahl, IBM Corporation

ABSTRACT

A first order characterization of a workload is given which permits the determination of an optimal storage, I/O and processor configuration. This analytic approach permits the handling of configurations of varying degrees of complexity.

The behavior of a computing system within a given workload environment is assumed to be a function of the three major variables: 1) sustainable I/O data rates in this environment; 2) storage or memory capacity; and 3) computer processing speeds attainable. Although programming support is indeed a major variable, it is assumed here that the characteristics of the programming is a function of the state of the art and is not a free variable for system configuration studies in the same sense as the others are. Consequently, the workload environment characterization is assumed to be made within the programming structure existing at a given point in time.

The workload characterization can be a very complex affair if one expects to account for anything but first order effects. For my purposes here, the first order effects are adequate to predict quite well the nature of the interaction of the three major variables in determining system behavior. The workload considered here is of a general purpose nature, consisting of a great many tasks ranging in size, running time and I/O requirements. A statistical characterization desired is the effect on I/O and processing requirements as a function of memory capacity allotted to the workload.

Two rather surprising observations have been made from gross global statistical data. The first of these is that the number of bits of I/O per instruction executed for a workload in its programming environment remains essentially constant. The amount of execution involved is not expected to alter appreciably, so the total amount of bits of I/O represented by the workload must therefore remain essentially constant. The second observation is that the amount of memory installed in a system is historically directly proportional to the speed of the processor. Recognizing that the record retrieval time from I/O has undergone minor change compared with processor speed change, one can deduce that twice the memory capacity permits execution of twice as many process steps per external record reference. And in conjunction with the first observation, these external records must be of approximately twice the average size.

For the current state of the art in programming there is apparently an upper limit on the size of records that may be dealt with as logical entities, and a corresponding upper limit to the amount of memory allotted to a single problem stream. Above this limit multi-programming is required to provide an adequate processing load for a high speed processor. Considering the multiprogramming situation as corresponding to "n" programs driving "n" I/O devices, we again observe an equivalent behavior where twice as much memory capacity permits twice as many process steps per external record reference by each program. The average record size summed over the collection of programs is again also twice as large.

With the above characterization of the workload it is possible to economically optimalize a system configuration to yield a minimum throughput capacity. For an approximate optimalization the memory and principal I/O devices can be optimalized together and the processing capacity optimalized in turn with the aggregation of memory and I/O. This procedure of local optimalization improves in validity as the costs of the I/O and memory become more extremely the most significant costs in the system.

The optimal amount of memory to associate with an I/O device should be that amount which minimizes the cost per bit of bringing in the workload. Assuming a simple I/O device with an access time of T, a transmission rate X, and a rental rate R. Correspondingly assume a memory with a rental rate per bit of r and the requirement of m bits of capacity to support a processor run of each bit of I/O, and let the total capacity be represented by M. The record size in bits will thus be M/m, the record reference time will be $T + \frac{M}{mx}$, the rental will be R + rM, and the bits retrieved will be M/m. The cost per bit retrieved is---

$$C/b(M) = \frac{(T + \frac{M}{mx})(R + rM)}{M/m}$$

The minimal cost per bit occurs when---

$$M_{OPT.} = \sqrt{\frac{TXRm}{r}}$$

The amount of memory assigned to the I/O device should then be equal to $M_{OPT.}$, unless this quantity exceeds the limit of programming support for single program streams. In that event the largest size applicable should be used.

More complex models of the I/O device, including arm motion, rotational latency and interference for arm motion may be employed. Similarly a mixture of record size limits can be assumed, where the available fraction of records which can still profitably be increased in size decreases with size. Alternatively the records can be continually increased in size but the number of useful bits retrieved could be considered as decreasing from M/m.

An even more complex model involving additional levels in the memory-I/O hierarchy can be analyzed rather roughly. From the two basic observations one can deduce that the probable time between references to a lower level of the hierarchy is inversely proportional to the capacity of this level and that the record size to be transmitted should be proportional to this capacity. If this latter requirement cannot be maintained at full usefulness of the retrieved information on this more global level, the storage hierarchy cannot be achieved at full economic benefit.

With the optimum choice of memory size to be employed with the principal I/O devices, the effective I/O performance and costs can be obtained. These must now be introduced into the determination of the processing speed requirements to optimalize the cost of performing the workload. From the first observation it is seen that each useful bit of I/O requires some number of instructions to be executed by the processor. If the time to process the useful contents of a record is too long in relation to the record retrieval time, the I/O and memory will not be fully utilized and the cost of processing the workload will increase. If the processor is too fast, the cost of the processor will become significant, without corresponding improvement in the rate at which the workload can be brought into the memory.

For the economic optimalization, let the processor performance, P, be measured in units of bits processed per unit time, where some fraction, f, of this processing cannot be overlapped with the record references. Assume that the cost of processing is linear such that ρ is the rental for one bit per unit time. The total time for processing the workload is then---

$$T_a + \frac{M}{mx} + \frac{fM}{m},$$

the rental is $R + rM + \rho P$, and the work done is still M/m. The cost per bit of processing is---

$$C/b(M, P) = \frac{(T_a + \frac{M}{mx} + \frac{fM}{mP})(R + rM + \rho P)}{M/m}$$

The minimal cost per bit occurs when---

$$P_{OPT.} = \sqrt{\frac{fmX(R+rM)}{\rho(T_a MX+M)}}$$

and the memory-I/O balance should be corrected to---

$$M_{OPT.} = \sqrt{\frac{T_a mXP(R+\rho P)}{r(fX+P)}}$$

The actual amount of system processing capability is obtained by providing the necessary multiplicity of such optimalized units. Again a more detailed optimalization can be carried out to accommodate a queuing model of multiple I/O devices served by a single processor or by multiple processors.

COMMUNICATIONS AND SYSTEM ARCHITECTURE

L. G. Roberts, Advanced Research Projects Agency

In order to continue to make significant progress during the next decade in the information processing field it is imperative that we be able to build on the totality of resources previously developed rather than rebuild everything for each new computer. More and more of our effort is being consumed in the duplication or translation of previous work. Also, since each center currently must store locally all the data it utilizes, our file storage requirements are growing exponentially. Thus the need and challenge is to find techniques which permit us to pool our software, hardware and data resources. Translators which permit the easy transfer of software and data between machines are helpful but do not stop the data storage explosion. Centralizing all computer activity onto one huge machine would solve the sharing problem but would drastically reduce the computing capability and flexibility available. A network of computers, on the other hand, does not inherently limit the flexibility or capability of the individual computer systems yet it permits very effective resource sharing.

The communication system developed for the ARPA Network provides sufficient reliability and responsiveness, at a reasonable cost, to make possible extremely effective resource sharing between geographically separated computer systems. The implication of resource sharing to the system architecture of the individual nodes in a computer network is currently being investigated. This paper describes some of the initial concepts which may affect computer system architecture in the future.

Currently, computer systems tend to concentrate all the resources required for the local users into one all-encompassing computer system. With an effective computer network, however, the system resources required to perform a given job need not be integrated into one system but can be distributed throughout the network. In general the services provided by a normal computer system can be broken into three major categories: terminal support, numerical processing and file services. Terminal support normally includes those functions which are closely tied to the user's interactive process such as editing, display maintenance and tablet character recognition. These terminal support services can often be most efficiently provided by a mini-processor located close to the user. The terminal support processor may either be directly connected to a large computer system or may access remote resources through the network. In the future I expect many organizations will find it more effective to have only a terminal support processor and obtain all their services from the network.

The numerical processing can be executed on whichever computer in the network is best suited for each job, whether by virtue of special purpose hardware, unique software or cost effectiveness. With an effective network the actual location where numerical services are performed is not particularly important, hence this function may be very widely distributed. However, it is not expected that a particular user will choose a different computer each day to perform the same job; more likely he will choose the best machine to begin with and change only when the economics warrant it.

Large data files will be stored and maintained in one particular network location since duplicate copies are both expensive and logistically complex to maintain. In many cases the data might be co-located or even integrated with numerical services, but often this is not required. One can envision an independent file processor with many levels of storage hierarchy, optimized to handle file retrieval, update and maintenance, including the capability to sort, merge and perform conditional queries. Such a file processor would not necessarily be associated with one particular numerical processing system but rather would service a distributed set of users having common interest in the data files. The effectiveness and economy of separating the file handling function from the numerical processing is not yet well understood, but it may well develop that this division of responsibility could lead to large improvements in efficiency through specialization.

The division of computer services into terminal support, numerical processing and file services is only the first glimpse of what may be an increasingly common and effective trend in computer system architecture made possible by the advent of computer networks. Although it has been possible to build very large computer hardware systems, the development of all-encompassing general purpose software monitors for large systems has been notably disappointing and at least for the foreseeable future it remains far more feasible and efficient to build specialized software systems perhaps handling a large number of users but within a restricted domain.

SYSTEMS ASPECTS OF STORAGE

J. E. Thornton, Advanced Design Laboratory, Control Data Corporation

DISTRIBUTION OF STORAGE

It is convenient and often necessary to distribute resources throughout a computing system. Especially in large computing systems the distribution of storage has a significant effect on the system performance. Given a little license about the nature of storage, one can find it literally everywhere within a system.

For example, storage elements exist both locally and remotely in the form of magnetic tape, magnetic disk, magnetic drum, magnetic core buffers, shift registers, delay lines, and high speed and low speed random access memories. The proper amount and location of each of these from a functional, organizational and a physical point of view have a bearing on the total system performance.

Some of the values of a de-centralized or distributed system of storage are listed below.

- Minimize high cost, high performance storage.
- Isolate failing elements for repair.
- Increase efficiency and cost effectiveness of information retrieval.
- Minimize data "traffic" on high cost communication lines.
- Minimize spurious accesses to bulk storage units.
- Increase the level of parallel operations in the whole system.

No one should underestimate, however, the job of distributing storage. An immediate hurdle is faced in deciding what to remove from the traditional domain of the Central Storage. The obvious advantage to this location is the access to the Central Processor with high efficiency. As a result, if any storage is removed, some processing must go with it. Now we are talking about distribution of processing as well.

This has its benefits, though. Independent, isolatable system elements offer a considerable advantage to system flexibility and performance. Add to that the value of physical isolation for maintenance and load balancing in a large system.

SPACE-TIME CONSIDERATIONS

The amount of storage can have a striking effect on the net performance of a computing system, whether measured as total job time or multiple job throughput. Often the presence of adequate storage at a particular point in a computing system will be a major factor in such measurements.

The choice of the size of a block of data to be moved within the storage hierarchy is affected by the storage units, by the job strategy, and by the system conventions and disciplines. Most rotating storage units tend to require large blocks of data in order to minimize waste of the storage medium in interblock gaps. However, some job accesses to storage only require a small amount of data.

To smooth over this inconsistency, an accepted strategy is to call for a reasonable block of data in each move through the system, in the hope that there will be further use of the whole block.

Particularly near the central storage, this leads to a high rate of activity and traffic with little assurance of effectiveness. A varying size element moving through the storage hierarchy can, however, produce a hopelessly complex set of tactics for the effective use of each level. Probably no major change is noticed in job time or throughput between the two schemes either.

A very effective match can be described between the Central Storage and an overflow or Extended Storage of lower cost in a multiprogrammed or time-shared environment. A fairly high degree of data traffic can be assumed to exist between the Central and the "overflow" in order to effectively handle a maximum number of simultaneous users. One advantage of this situation is that it lends itself to queueing techniques, thereby allowing a rotating device as an overflow unit.

VIRTUAL MEMORY

A scheme introduced a few years ago, called Virtual Memory, has great potential for handling multiple levels of storage. To date the methods of utilizing "extended" storage and "back-up" storage have been primitive and marginally effective. It is not always possible to organize a job or the data in a manner which is optimum for each level. Since this can be a major cause of degraded system performance, any improvement would have great value.

The virtual memory scheme tends to remove the multiple levels of storage from the view of the user program. It assumes that the system can do a better job of controlling storage levels. This does not eliminate the possibility of hints and advice from the user program to the system.

It has the additional tremendous advantage of eliminating direct Input-Output operation from the user program. Instead, data can be transferred throughout a system from storage to storage in forms suitable to that point in the system. This data can be staged, buffered, compressed, and so on with no I/O involvement except right at the I/O device itself. While virtual memory, which is a central processor consideration, is not responsible for all of this handling, it is a key part of it.

A particular advantage of some virtual memory implementations is that programs may vary in size with no constraints except performance changes due to operating on different levels of storage. This advantage allows for system growth and also allows for networks of computers with different amounts of storage at each storage level.

SYSTEM CONTROL OF STORAGE

Typically, the control of all levels of storage has been explicitly in the hands of the user program. Since large computing systems have changed from single-owner dedicated systems to shared service centers with multiple users, the explicit control is breached. The owner, who controls the system programs, must now intervene in the handling of all levels of storage in the interests of efficiency, load balancing, security and maintenance.

Typical control of all levels of storage by the owner's system has fallen to the central processor with a considerable impact on its performance and on the amounts of central storage taken over for this purpose. Control tables, directories, buffer areas and so on are large and getting larger.

It is not at all obvious that this is in fact the optimum method of system control of storage. In fact the advantages claimed for distributed storage in a computing system can be equally claimed for distributed control. Storage subsystems which are optimized around the specific storage device and which obey system discipline and conventions can provide a number of advantages, among them:

- Increased total system performance,
- Isolation in case of breakdown,
- Flexible growth,
- Optimum introduction of new storage technology.

For the computing system to take advantage of such subsystems, the system structure, or scaffolding, depends on clean interface definitions. Since these interfaces are of a high level nature, one must presume a system made up of "intelligent" units, with high level message channels interconnecting them. The use of peripheral processors, mini-computers or programmable terminals is, of course, obvious here.

System elements in this kind of system should be designed to accomplish a maximum amount of the function assigned to them without any help from another element. This represents a base-line of systems control over the storage within that element. The total system builds on these system elements with higher level control being accomplished by high level message interaction between elements. Data is passed around through the system in a disciplined way including methods of identifying, protecting and backing it up.

It is clear that traditional explicit methods of locating and positioning data in the storage hierarchy are now defunct. The system designer must now develop the discipline to control storage by implicit methods as seen from the user program.

AUTHOR INDEX

Amano, K., 43
Amdahl, G. M., 371
Amdahl, L. D., 1
Antonetti, V. W., 30
Arnold, R. F., 345
Austin, D. C., 141
Bartlett, J. P., 299
Belshe, R. A., 126
Berg, R.O., 336
Berny, J. G., 159
Blumenthal, E. I., 343
Bourgoin, J. P., 159
Broadbent, B. D., 73, 352
Carter, W. C., 229
Chapman, D. W., 56
Cheek, T., 176
Chu, R. C., 30
Clark, N., 178
Clinnick, M. L., 126
Cogar, G. R., 9
Culler, G., 39
Darr, J. F., 115
Donzel, A. A., 159
Ellis, T. O., 38
Fechter, K. R., 180
Fedde, G. A., 103
Finch, T. R., 93
Flinders, M., 314
Frank, W. L., 7
Fulmer, L. C., 325
Gardner, P. L., 314
Geffner, S. L., 278
Giloi, W., 165
Graf, P., 83
Granberg, M. L., 180
Harding, P. A., 100
Haring, D. R., 186
Hobbs, L. C., 5
Hodges, D. A., 96
Hornreich, R., 64
Jack, R. W., 357
Jefferson, R. F., 120
Jessep, D. C., Jr., 229
Johnson, M. D., 336
Johnston, R. F., 89
Jones, A. L., 195
Kada, A., 43
Kamibayashi, T., 43
Kawarada, H., 261
Kay, A. G., 290
Kobayashi, T., 43
Komazawa, Y., 43
Komuro, K., 43
Kriz, T. A., 250
Leininger, J. C., 307
Llewelyn, R. J., 314
Lloyd, R. H., 354
Marcus, R. S., 109
Marinos, P. N., 240
Matcovich, T. J., 13
Meilander, W. G., 325
Meissner, L. P., 126
Minshull, J. F., 314
Moore, G. E., 107
Moreno, E., 175
Oshima, S., 43
Ossanna, J. F., 40
Pascuzzo, A. L., 30
Perlis, H. W., 205
Raisanen, W. R., 105
Reddy, D. R., 282
Reitz, G., 189
Roberts, L., 373
Ross, J. P., 2
Roth, J. P., 247
Rubinstein, H., 64
Sahni, R. J., 213
Sato, T., 261
Seeds, R. B., 10
Seely, J. H., 30
Selander, J. H., 171
Siemens, A. G., 83
Simon, W. F., 31
Skubiak, M. T., 30
Slattery, L. J., 94
Slotnick, D. L., 367
Soller, T. M., 180
Strand, E. M., 205
Sutherland, I. E., 37
Teixeira, J., 64
Tewlow, J. S., 109
Thornton, J. E., 374
Tickell, R., 355
Urano, Y., 220
Vezza, A., 148
Vickers, D. L., 270
Wadia, A., 229
Wadsworth, D. J., 30
Watanabe, T., 220
Wieselman, I. L., 360
Wilcox, D. L., 20
Winningstad, C. N., 141
Winson, S. E., 205
Yamamoto, H., 220
Yang, H. S., 363
Yoshiyama, M., 261

EXHIBITORS

ASSOCIATION FOR COMPUTING MACHINERY, INC.
1133 Ave. of the Americas
New York, N.Y. 10036

ADVANCED SPACE AGE PRODUCTS, INC.
4308 Wheeler Ave.
Alexandria, Va. 22304

AFIPS PRESS
210 Summit Ave.
Montvale, N.J. 07645

ATLANTIC TECHNOLOGY CORP.
Somers Point, N.J. 08244

AUERBACH INFO, INC.
121 N. Broad St.
Philadelphia, Pa. 19107

BEEHIVE MEDICAL ELECTRONICS
1473 S. 6th West
Salt Lake City, Utah 84104

CIPHER DATA PRODUCTS
7545 Convoy Ct.
San Diego, Calif. 92111

CIT - COMPAGNIE INDUSTRIELLE DES TELECOMMUNICATIONS
33, Rue Emeriau
Paris 15, France

COMPACE CORP.
7413 Washington Ave., So.
Edina, Minn. 55435

COMPUTER COMMUNICATIONS, INC.
701 W. Manchester Blvd.
Inglewood, Calif. 90301

COMPUTER CONSOLES, INC.
1257 University Ave.
Rochester, N.Y. 14607

COMPUTER DESIGN PUBLISHING CORP.
Baker Ave.
W. Concord, Mass. 01781

COMPUTER DIGITAL SYSTEMS, INC.
Logan Square
Norristown, Pa. 19401

COMPUTER PERIPHERALS CORP.
5037 Ruffner St.
San Diego, Calif. 92111

COMPUTER SCIENCES CORP.
6565 Arlington Blvd.
Falls Church, Va. 22046

CONFERENCE BOOK SERVICE, INC.
705 Prince St.
Alexandria, Va. 22314

CONNECTICUT TECHNICAL CORP.
3000 Main St.
Hartford, Conn. 06120

CONVERSATIONAL COMPUTING, INC.
10465 Melody Dr., #312
Denver, Colo. 80234

CULLER/HARRISON, INC.
5770 Thornwood Dr.
Goleta, Calif. 93017

DATA DISC, INC.
1275 California Ave.
Palo Alto, Calif. 94304

DATARAM CORP.
Rt. 206
Princeton, N.J. 08540

DATAWEST CORP.
7503 East Osborn Rd.
Scottsdale, Ariz. 85252

DELTA DATA SYSTEMS
Woodhaven Industrial Park
Cornwells Heights, Pa. 19020

DOCUMATION INC.
841 E. New Haven Ave.
Melbourne, Fla. 32901

DSI SYSTEMS INC.
11810 Parklawn Dr.
Rockville, Md. 20852

EASTMAN KODAK CO.
343 State St.
Rochester, N.Y. 14650

EDP NEWS SERVICES, INC.
514 10th St., N.W.
Washington, D.C. 20004

EDWIN INDUSTRIES CORP.
11933 Tech Rd.
Silver Spring, Md. 20904

ELECTRONIC LABORATORIES INC.
3726 Dacoma St.
Houston, Tex. 77018

GLOBAL MARKETING PROGRAM
Bureau of International Commerce
U.S. Dept. of Commerce
Washington, D.C. 20230

GOVERNMENT EXECUTIVE MAGAZINE
326 Shoreham Bldg.
Washington, D.C. 20005

GRAPHIC DATA, INC.
169 Bedford St.
Burlington, Mass. 01803

GRAPHIC SCIENCES, INC.
Corporate Dr., Commerce Park
Danbury, Conn. 06810

HOUSTON INSTRUMENTS, DIV. OF BAUSCH & LOMB
4950 Terminal Ave.
Bellaire, Tex. 77402

IBM CORPORATION
18100 Frederick Pike
Gaithersburg, Md. 20760

INFORMATION WEEK
288 Park Ave., W.
Elmhurst, Ill. 60126

INTERNATIONAL DATA SCIENCES, INC.
100 Nashua St.
Providence, R.I. 02904

LOCKHEED ELECTRONICS CO.
6201 E. Randolph St.
Los Angeles, Calif. 90022

LOGABAX, S.A.
79, Av. Aristide Briand
94 Arcueil, Paris, France

MEAD (DATA) CENTRAL CORP.
1254 Jefferson Davis Highway
Arlington, Va. 22202

MECHANICAL ENTERPRISES INC.
5249 Duke St., Ste. A
Alexandria, Va. 22304

MONITOR DISPLAYS, an Aydin Co.
401 Commerce Dr.
Fort Washington, Pa. 19034

ON-LINE COMPUTER CORP.
370 Ludlow St.
Stamford, Conn. 06902

PHOTO DATA, INC.
419 7th St., N.W., Ste. 500
Washington, D.C. 20004

PRENTICE-HALL, INC.
Englewood Cliffs, N.J. 07632

SAGEM
6, Avenue d'lena
75, Paris 16, France

SEALECTRO CORP.
225 Hoyt St.
Mamaroneck, N.Y. 10543

TIME SHARE PERIPHERALS CORP.
P.O. Box 361
Wilton, Conn. 06897

TRANSMISSION MEASUREMENTS INC.
1051 Clinton St.
Buffalo, N.Y. 14206

TRI-DATA CORP.
800 Maude Ave.
Mountain View, Calif. 94040

TYCO LABORATORIES, INC.
Digital Devices Div.
200 Michael Dr.
Syosset, N.Y. 11791

VISICON, INC.
P.O. Box 1008
State College, Pa. 16801

WANG LABORATORIES, INC.
836 North St.
North Tewksbury, Mass. 01876

JOHN WILEY & SONS, INC.
605 Third Ave.
New York, N.Y. 10016

1970 IEEE INTERNATIONAL COMPUTER GROUP CONFERENCE AND EXPOSITION

STEERING COMMITTEE

Conference Chairman Bob O. Evans
IBM Corporation

Conference Vice Chairman Donald E. Doll
IBM Corporation

Technical Program Committee

Chairman Dr. Gardiner L. Tucker
Department of Defense

Co-Chairman Dr. Martin Greenberger
The Johns Hopkins University

Co-Chairman Maurice Allegre
Office of the Prime Minister, Republic of France

Exhibits Committee

Chairman Leland D. Whitelock
Naval Ship Engineering Center

Vice Chairman Robert C. Carpenter
Naval Ship Engineering Center

Local Arrangements Committee

Chairman Murray Stein
International Computing Company

Vice Chairman Paul I. Hazan
Link Division, Singer Company

Finance Committee

Chairman Henry S. Forrest
CDC

Vice Chairman Harry Hayman
NASA

Publications Committee

Chairman Margaret R. Fox
National Bureau of Standards

Vice Chairman Thomas C. Bagg
National Bureau of Standards

Committee Member Robert Tietler
West and Brady

Publicity and Public Relations Committee

Chairman Theodore F. Watts
Mech. Enterprise, Inc.

Vice Chairman Andrew J. Cella
IBM Corporation

Co-Chairman John A. Camuso
Computer Design

Registration Committee

Chairman John E. Voyles
Informatics, Inc.

Vice Chairman James A. Cunningham
ESSA

Special Activities Committee

Chairman L. Jean Worley
Harry Diamond Labs.

Vice Chairman Mrs. Judith Roseman

Student Committee

Chairman Dr. Andrew G. Favret
Catholic University

Vice Chairman Dr. Richard H. Austing
University of Maryland

Technology Theater Committee

Chairman A. Eugene Miller
Auerbach Corporation

Vice Chairman Francis M. Blake
Auerbach Corporation

Secretary to Steering Committee

Mrs. Barbara T. Watson
IBM Corporation